SQUARE ENIX

Natural Expression
自然な表現を目指す

株式会社スクウェア・エニックス
研究開発部

講演内容

- Natural Expressionとは

  開発ディレクター

  藤井　栄治

  - GPU Global Illumination Renderer

    シニア・ソフトウェア・デザイン・エンジニア

    大垣　真二

  - 音声認識アプローチ

    テクニカル・スーパーバイザー

    井上　仁

    ソフトウェア・デザイン・エンジニア

    清原　理

Natural Expressionとは

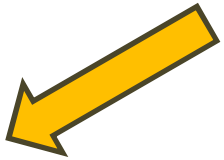
Real is not **Natural**

- Real

  —— *a.* 真の, 本物の; 実在[存]する, 現実の

- Natural

  —— *a.* 自然[天然]の, 自然界の; 加工していない

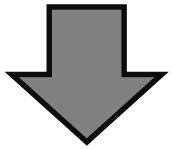


加工していない！

加工している例

COMPOSITE

多重イメージ・レイヤーによるコンポジット

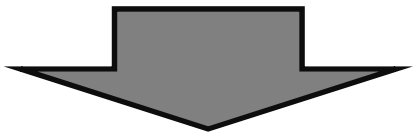


人の目に頼るワークフロー

自然な感じに見せるには、職人技が必要

現在のCG制作

- モーションキャプチャーデーターの調整
- クロスシミュレーション後の調整
- 手描きのテクスチャーマップ（写真加工）
- 3Dペイントによるディスプレイスメントマップ
- その他。。。。。。



結果的に職人の手を多く必要としている

計算が重いため、もしくは
何回も計算や、やり直しをすると
時間が間に合わないため

多くの優秀な職人が必要で、満たない場合

自然な感じは、失われてしまう

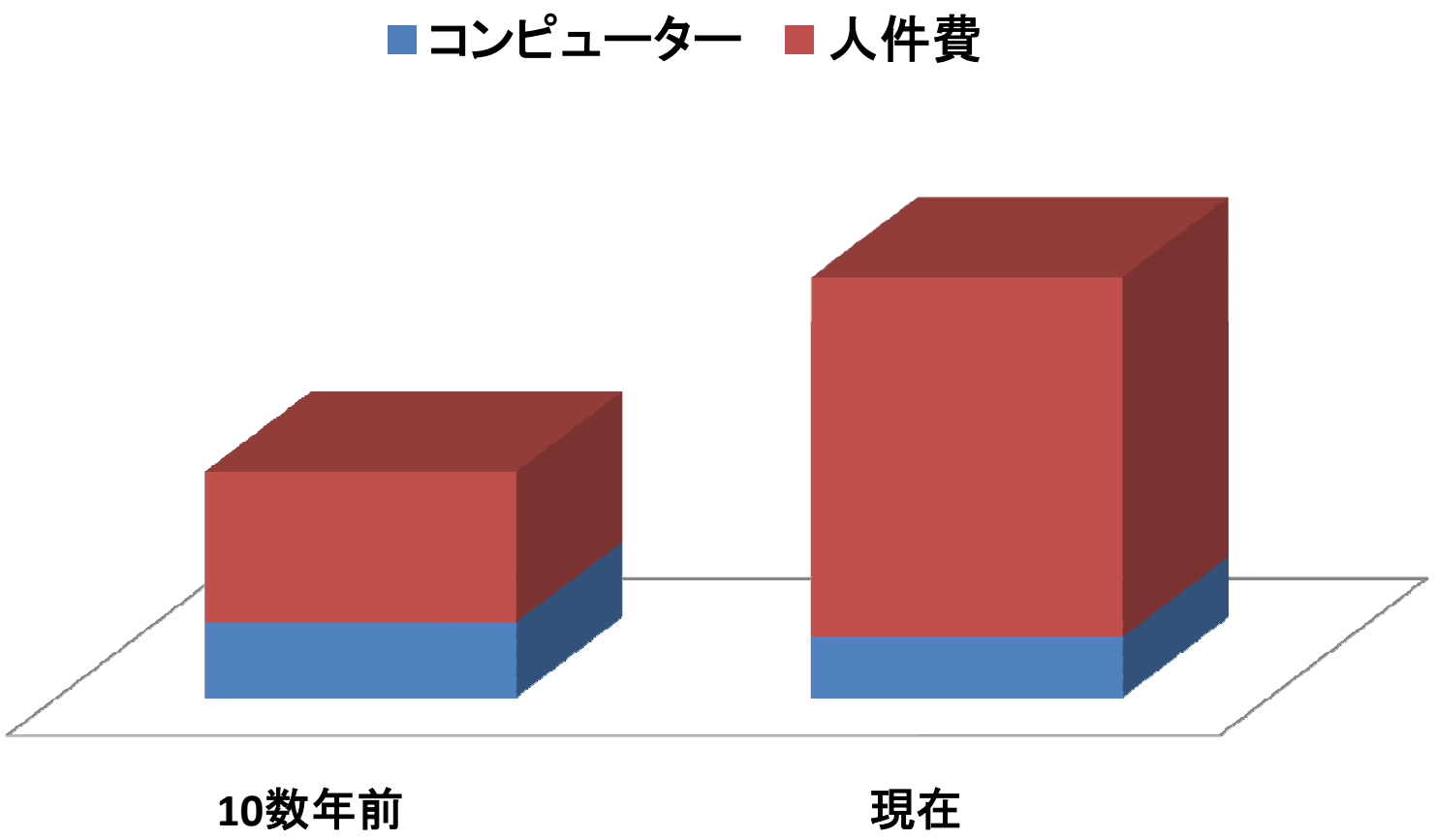
職人は多くいないし、多く雇えない

現在と過去のコスト比較

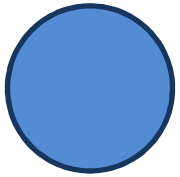
- １０数年前
  コンピューターは、シリコングラフィック社製が多く、
  ソフト込みで1台１０００万円前後のコスト

- 現在
  コンピューターは、パーソナルコンピューター
  （WINDOWS）が多く、1台８０万円前後のコスト
  ソフト代は２００万円前後
  作業する人は、約3倍増えた

10数年前と現在のコスト比較



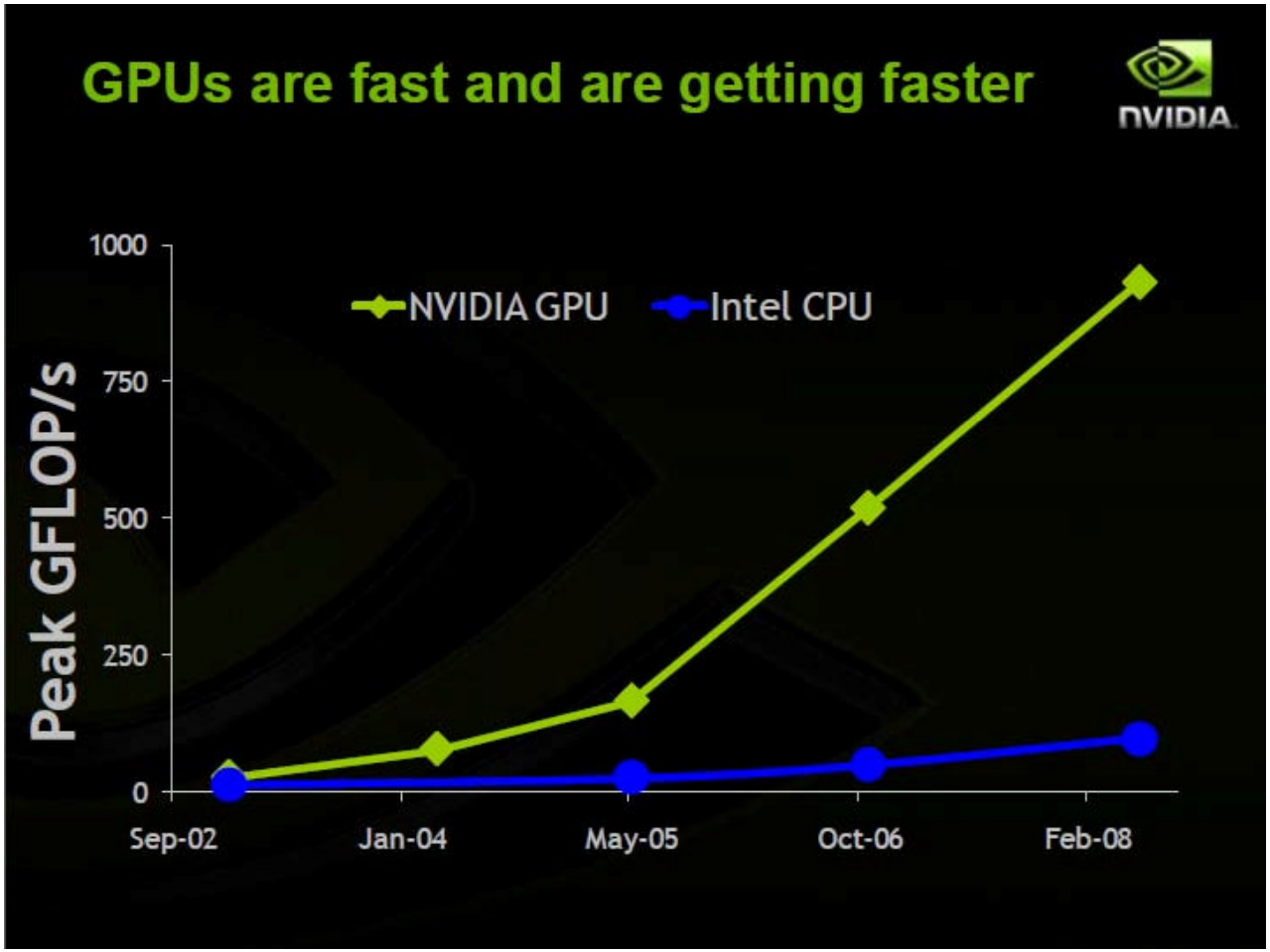
コンピューターの性能、性質を駆使した取組

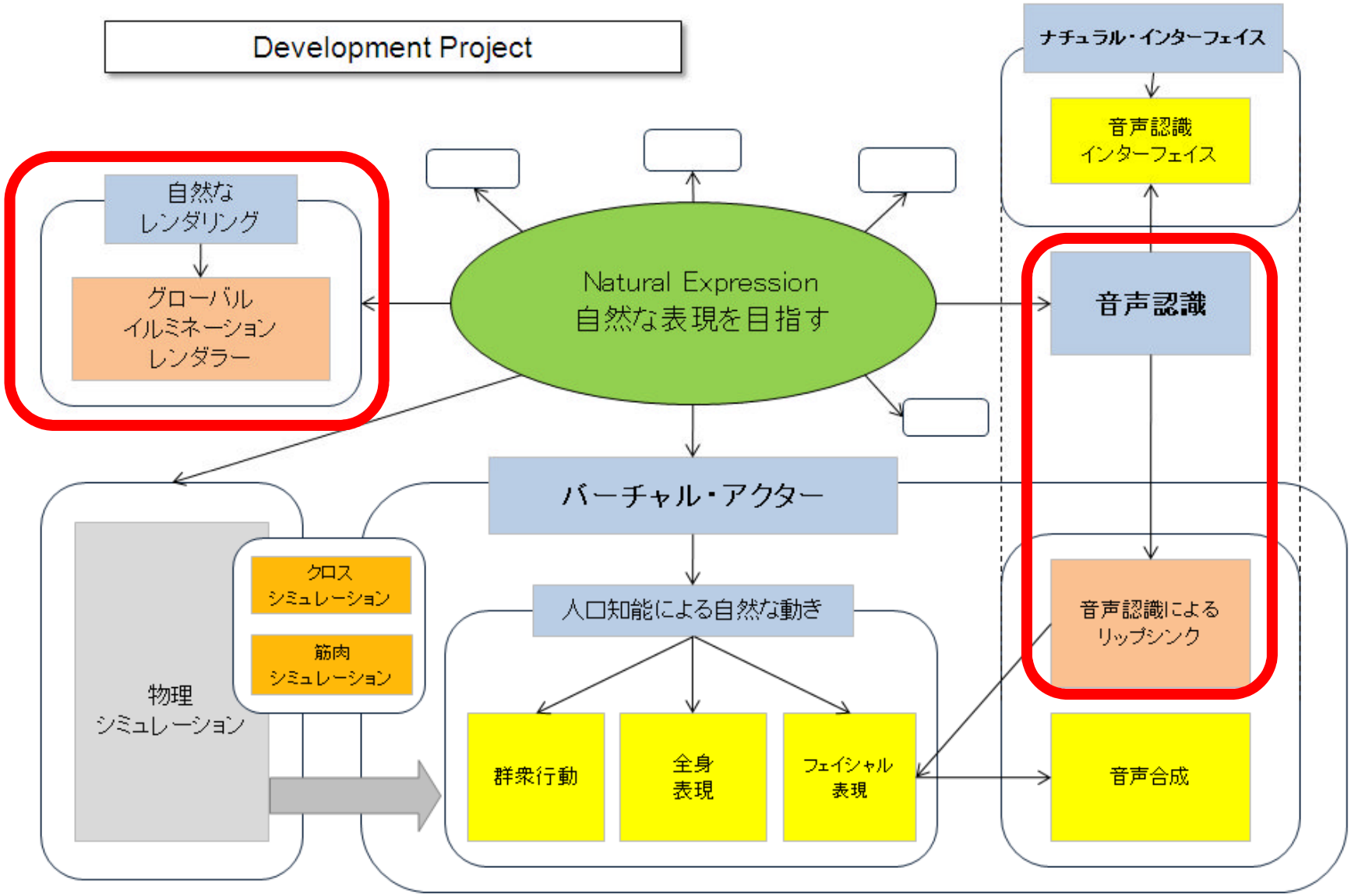


- 高速
- 自然な計算
- 自動化
- プロシージャル化

- キーフレームアニメーション
- 手描きのテクスチャー
- 手修正
- 遅い計算

CPU vs. GPU


GPUs are fast and are getting faster
NVIDIA
Peak GFLOP/s
1000
750
500
250
0
NVIDIA GPU
Intel CPU
Sep-02
Jan-04
May-05
Oct-06
Feb-08



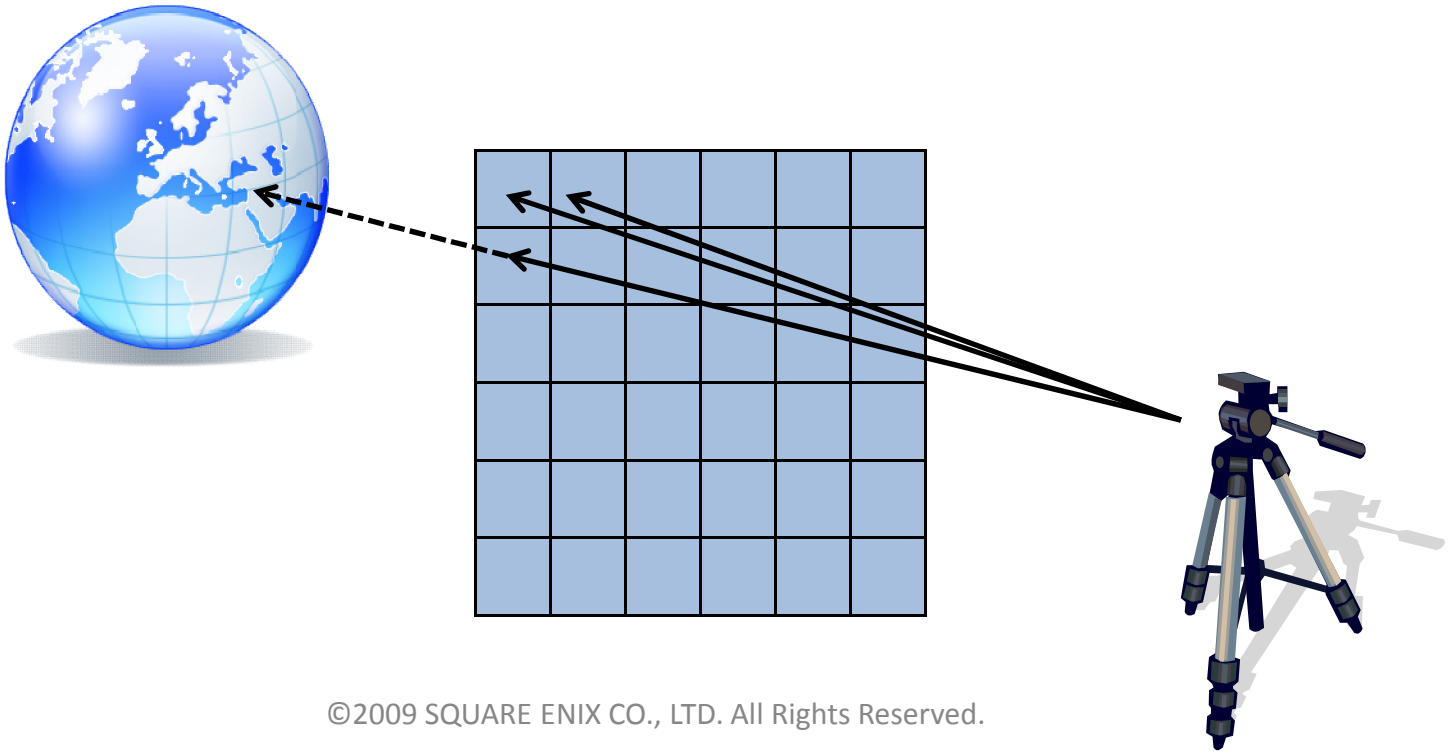
GPU Global Illumination Renderer

内容

- Global Illuminationとは?
- 我々の目標
- システム概要
- 現在の開発状況
- 技術紹介

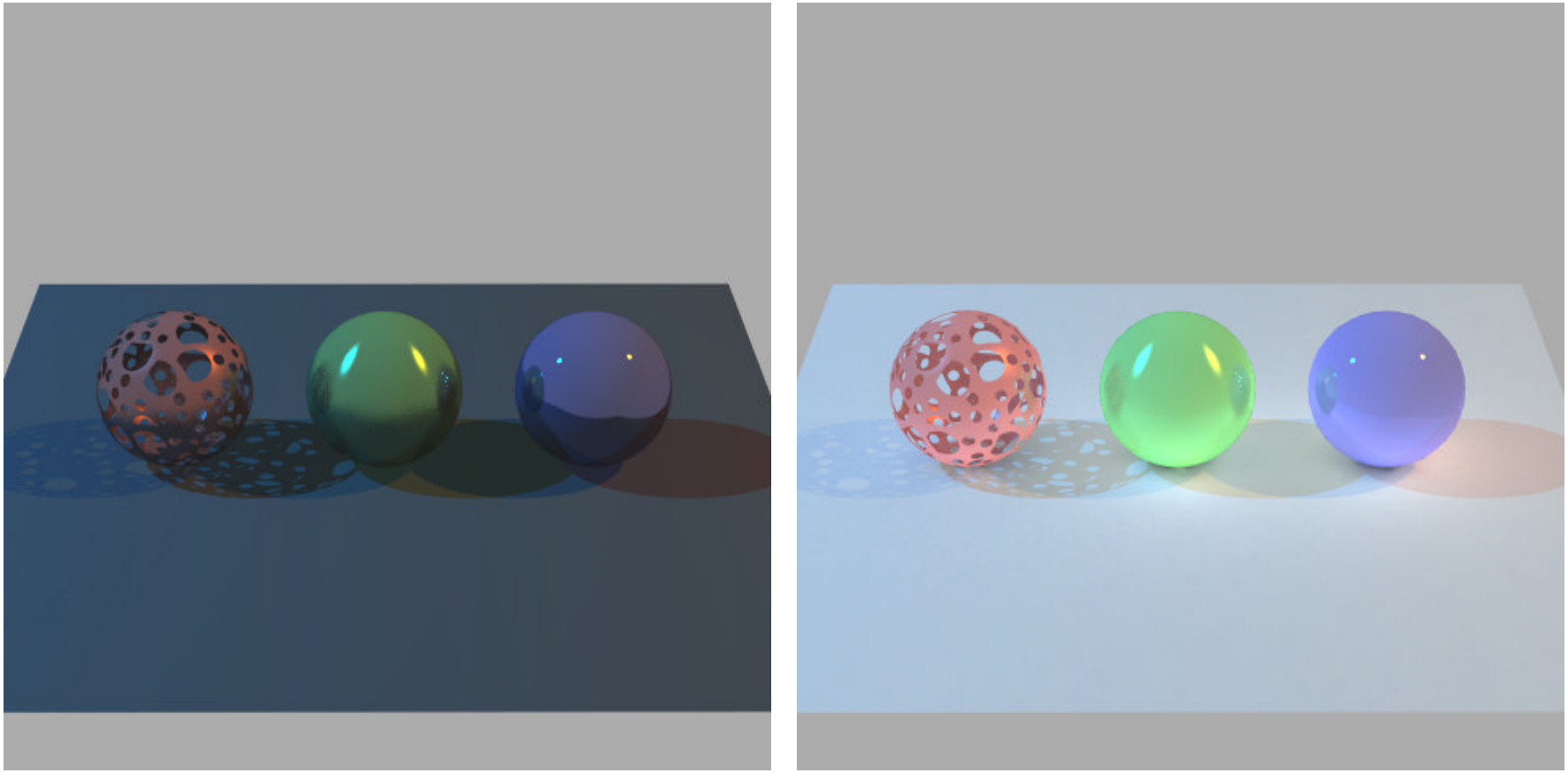
レイ･トレーシング

- ピクセルを通る光線を追跡して、画像を生成する方法



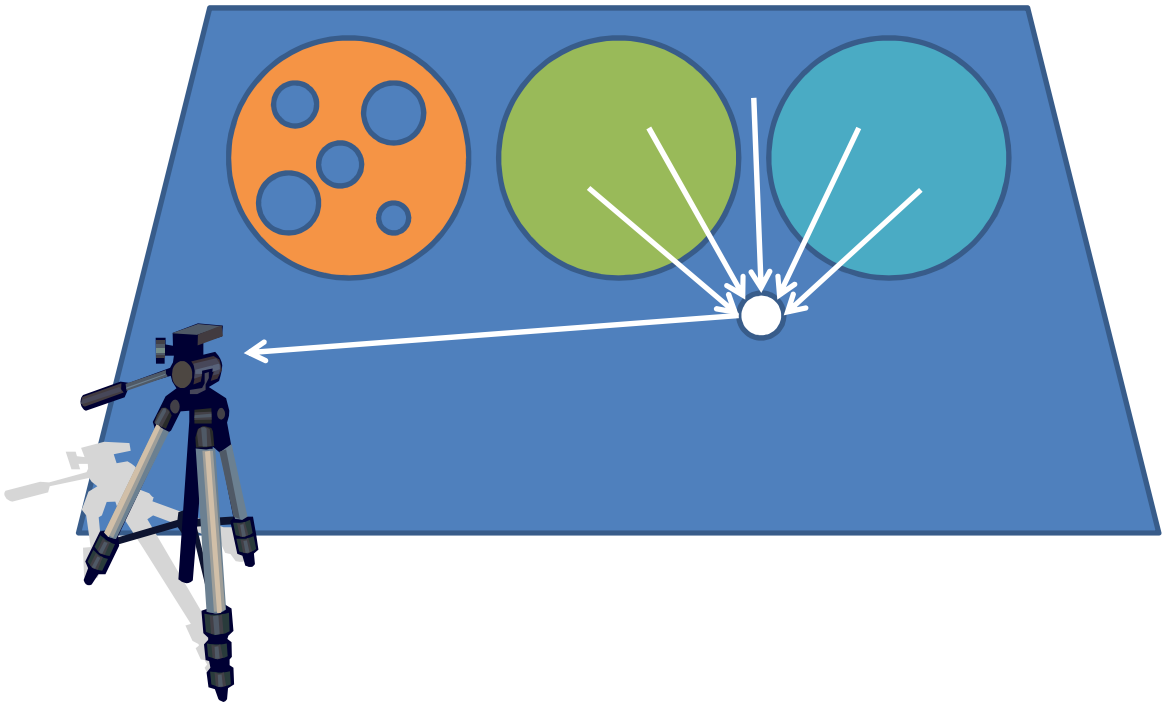
Global Illumination

- 直接照明だけではなく、間接照明も考慮する



Global Illumination

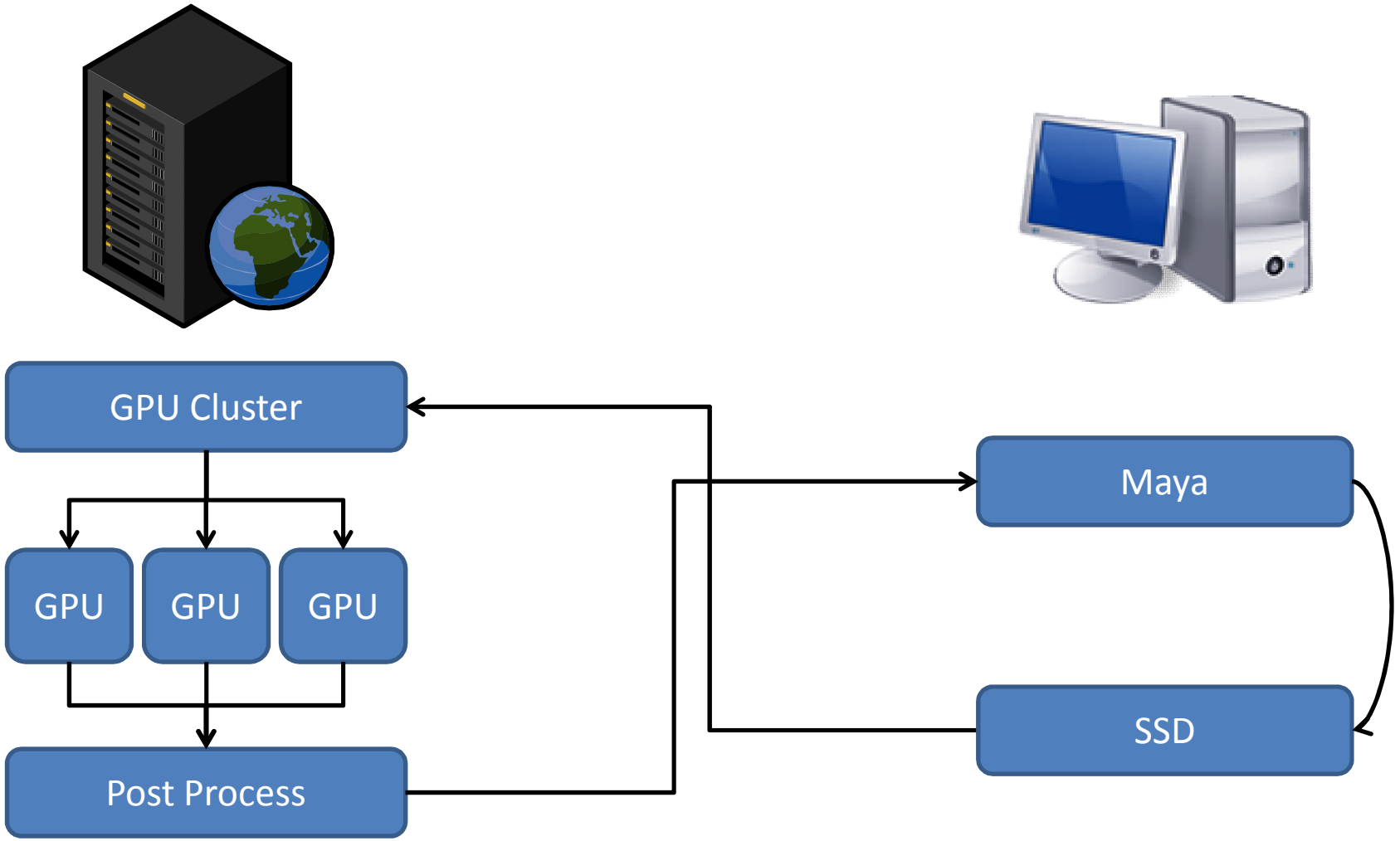
- 光はすべての方向からやってくる



我々の目標

- 速い
  - *全ての処理はGPUで行われなければならない*
- リアル
  - *物理的に正しい*

システム概要



現在の開発状況

機能	実装状況
Ray-Triangle Intersection (N3-Tree as an Accelerator)	○
Basic Materials (Reflection, Refraction, Emission, etc.)	○
Texture Mapping	○
Final Gathering	○
Fur/Hair	○
Direct Ray Tracing of Parametric Surfaces (such as Trimmed NURBS)	△
Photon Mapping	△
Volume Rendering	×
Motion Blur	×
Procedural Texture	×
Displacement Mapping	×

○実装済み　△ほぼ実装済み　×実装中

現在の開発状況

CPU/GPU	スピード
Intel Xeon X5355 x 2(8コア)	**1**
NVIDIA Quadro FX5800	**6**
NVIDIA Quadro Plex 2200 D2	**12**

技術紹介

- どのような技術が使われているか?
  - パラメトリック曲面の直接レンダリング
  - 幅優先レイ・トレーシングとストリーム・フィルタ
  - フラグメント・コレクション
  - キャッシュ・オブリビアス・データ・レイアウト
  - コヒーレント・レイ・トレーシング

パラメトリック曲面の直接レンダリング

- Trimmed NURBSなど
- データ･サイズの削減

幅優先レイ･トレーシング

- レンダリング方程式のノイマン級数展開

$$L_{out}(x,\vec{\omega}_{out}) = \int_{\Omega} f_r(x,\vec{\omega}_{out},\vec{\omega}_{in}) L_{in}(x,\vec{\omega}_{in})(\vec{\omega}_{in},\vec{n}) d\vec{\omega}_{in}$$

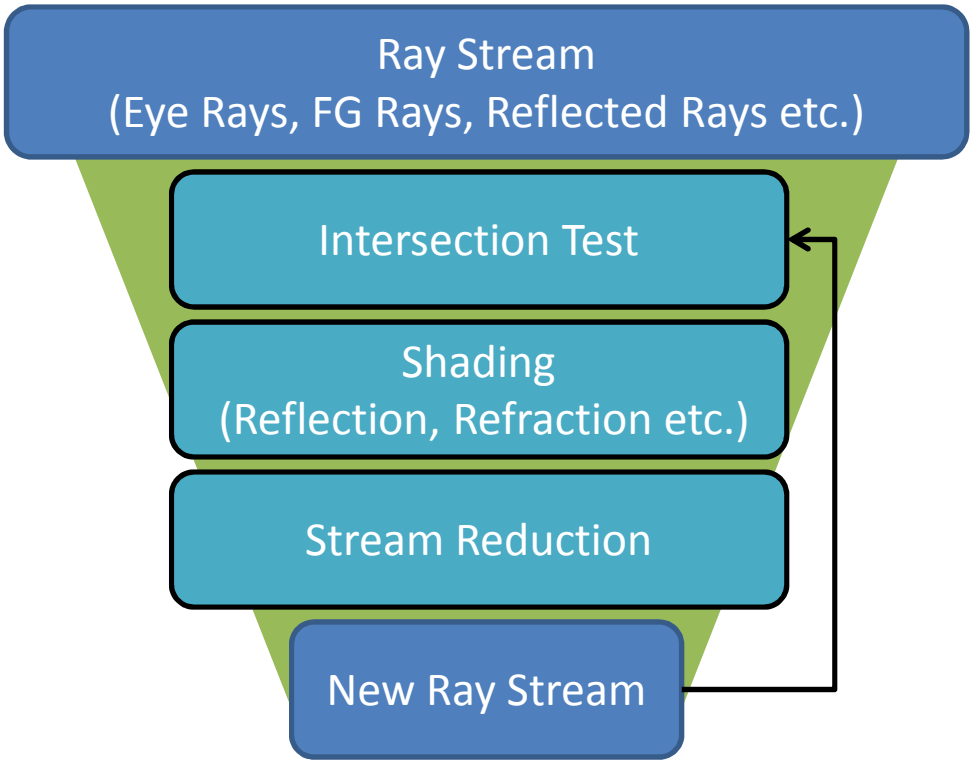
$$L = L_e + TL$$

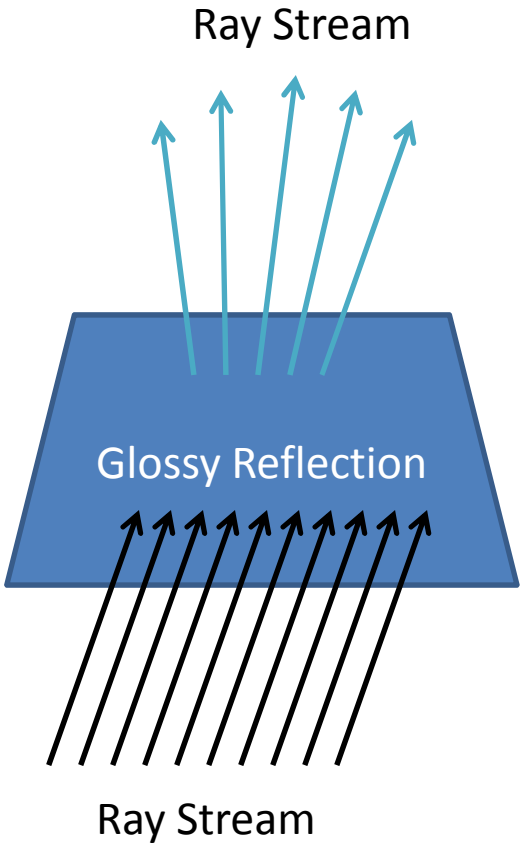
$$< Tg > (x,\vec{\omega}_{out}) = \int_{\Omega} f_r(x,\vec{\omega}_{out},\vec{\omega}_{in}) g(x,\vec{\omega}_{in})(\vec{\omega}_{in},\vec{n}) d\vec{\omega}_{in}$$

$$L = L_e + TL_e + T^2 L_e + T^3 L_e + T^4 L_e + \ldots$$

幅優先レイ・トレーシング

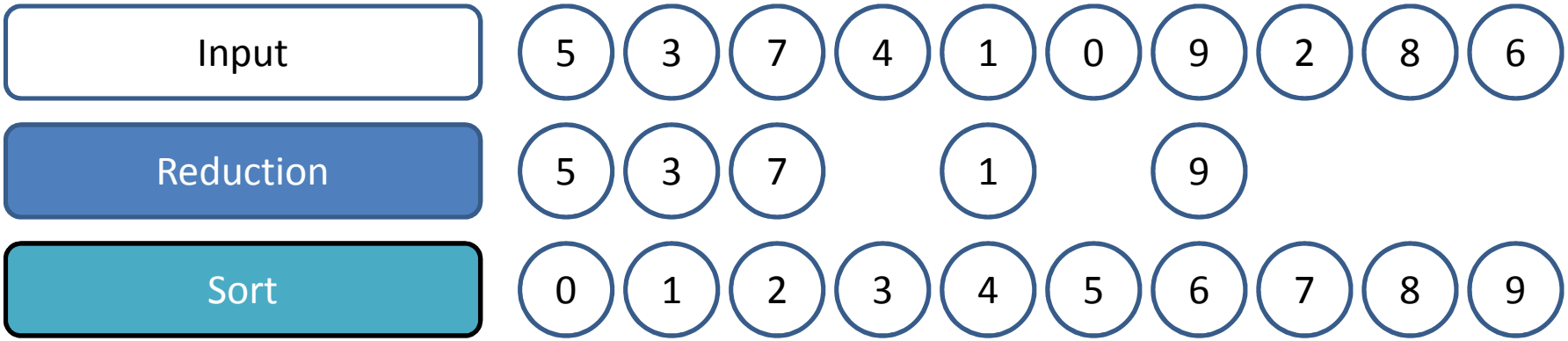
- MPP (Massively Parallel Processors)を最大限利用する



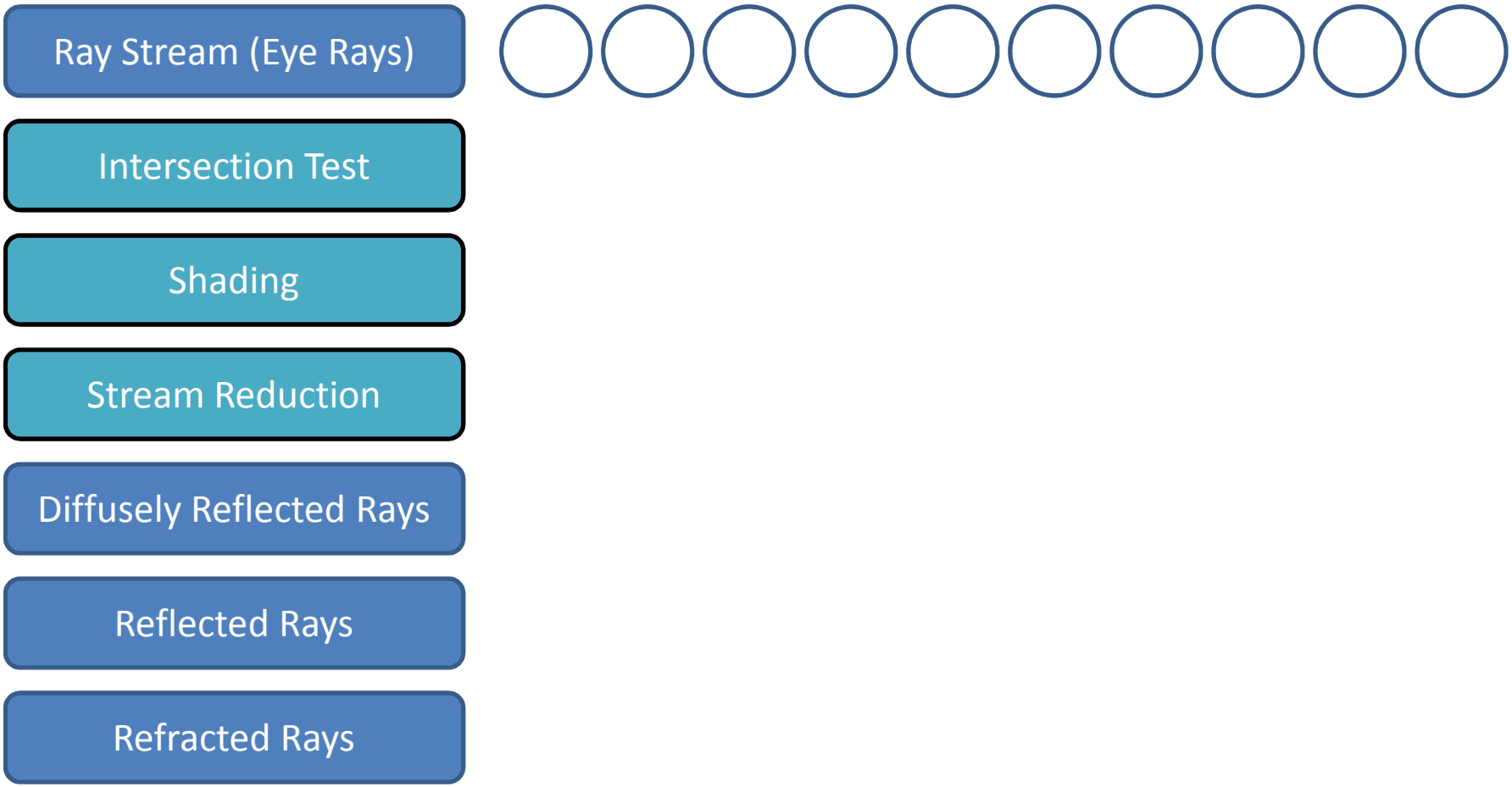


ストリーム・フィルタ

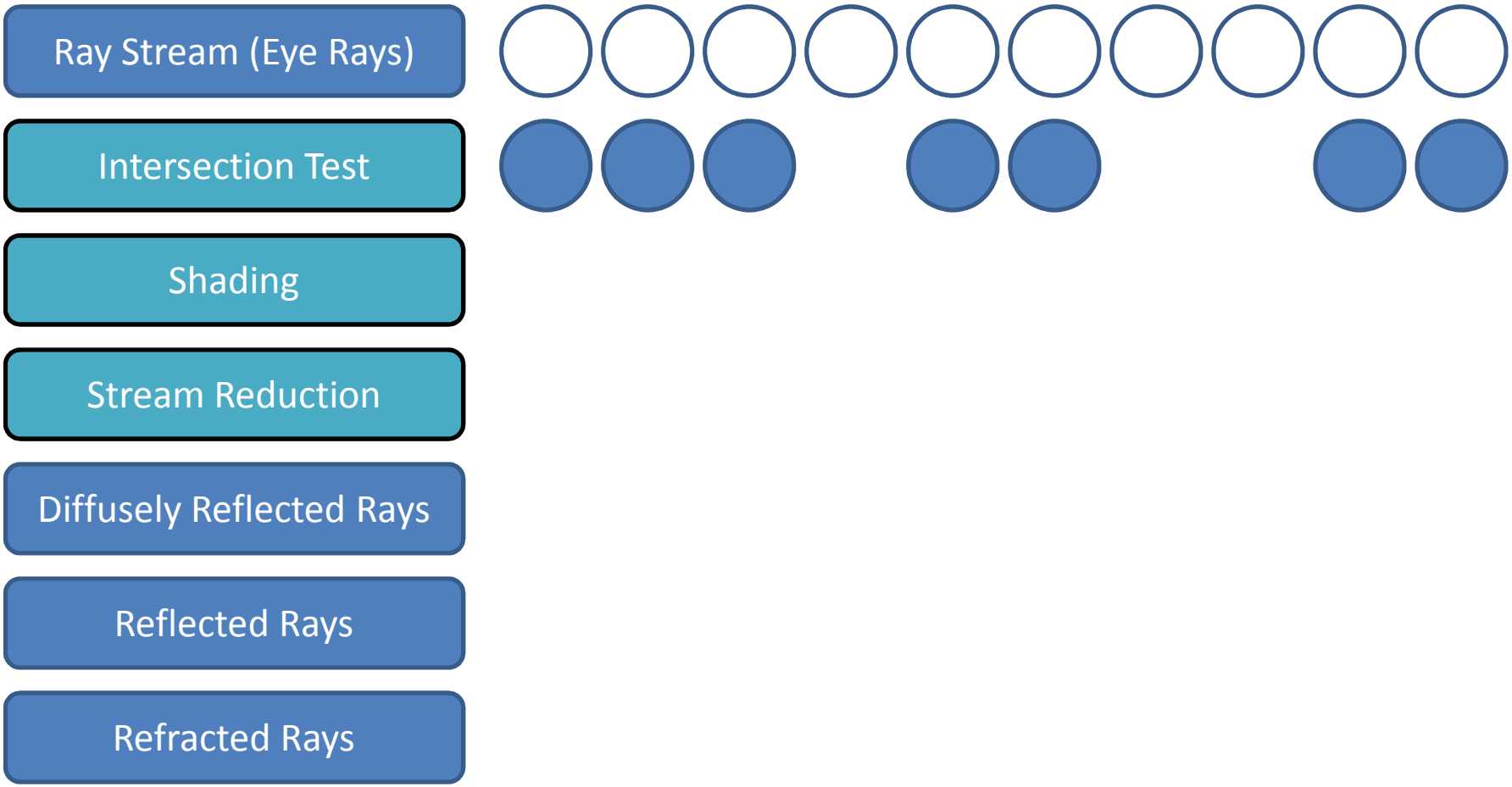
- “Coherent Ray Tracing via Stream Filtering” (Christiaan P. Gribble et al.)



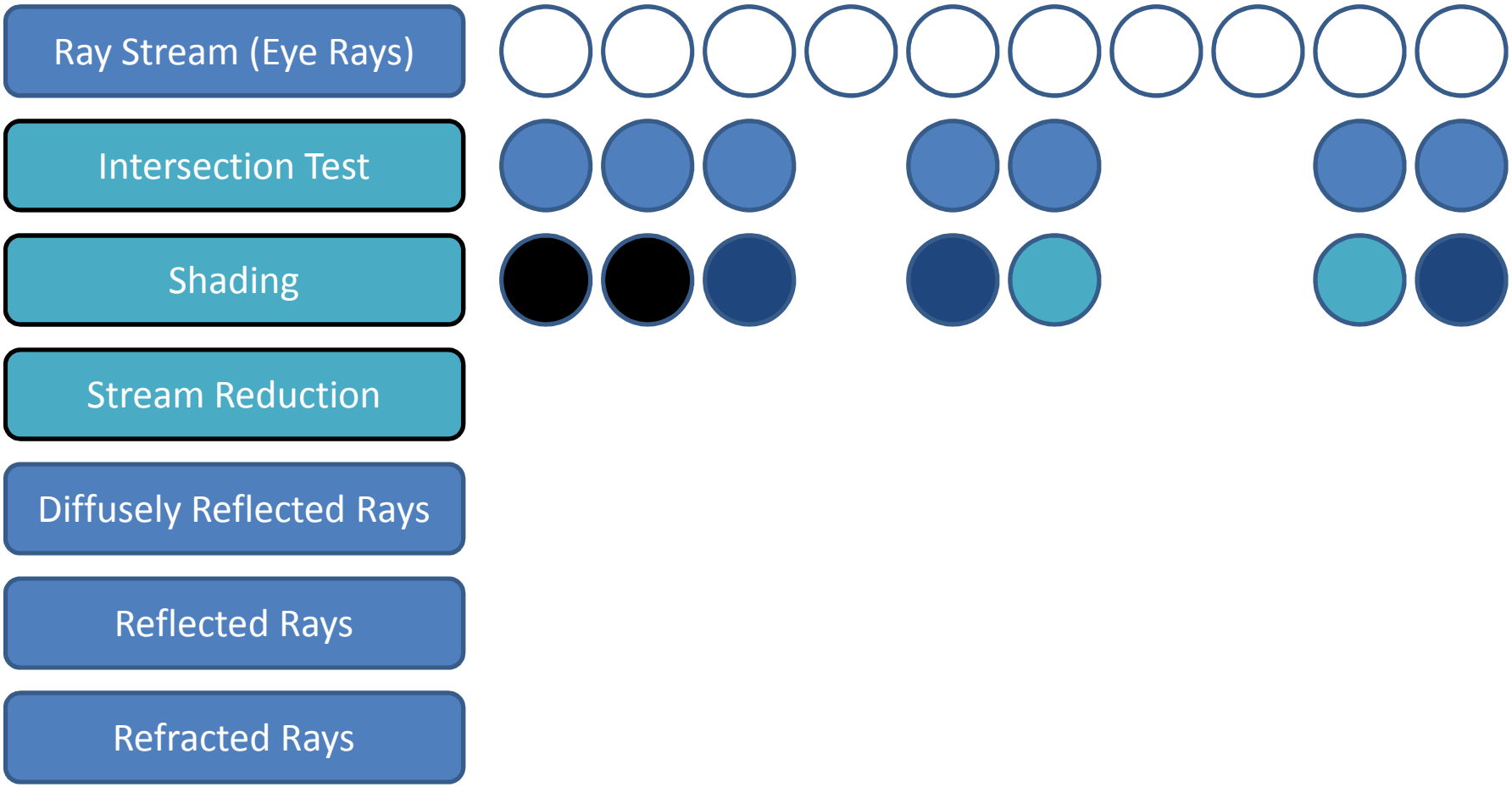
ストリーム・リダクション



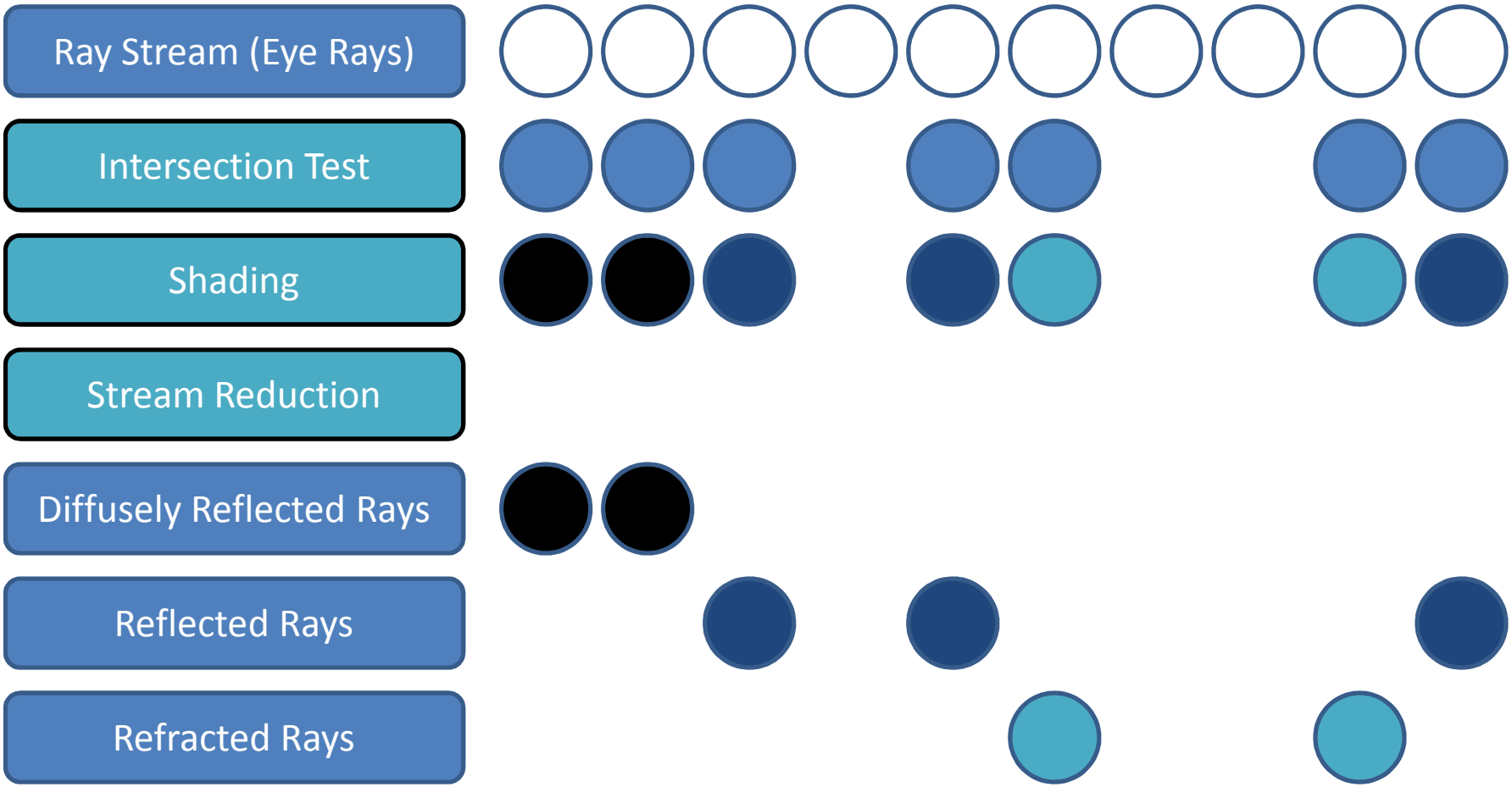
ストリーム・リダクション



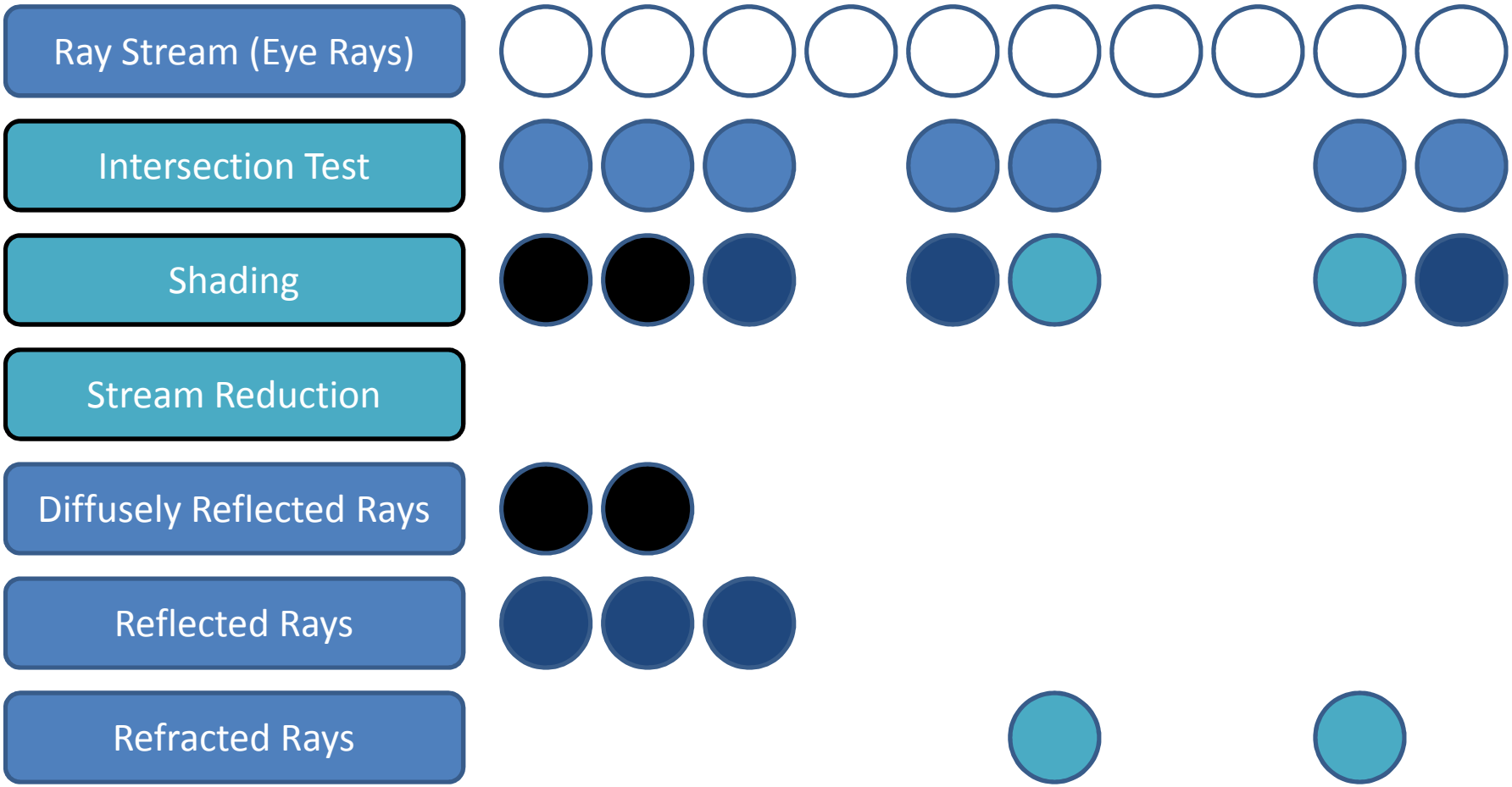
ストリーム・リダクション



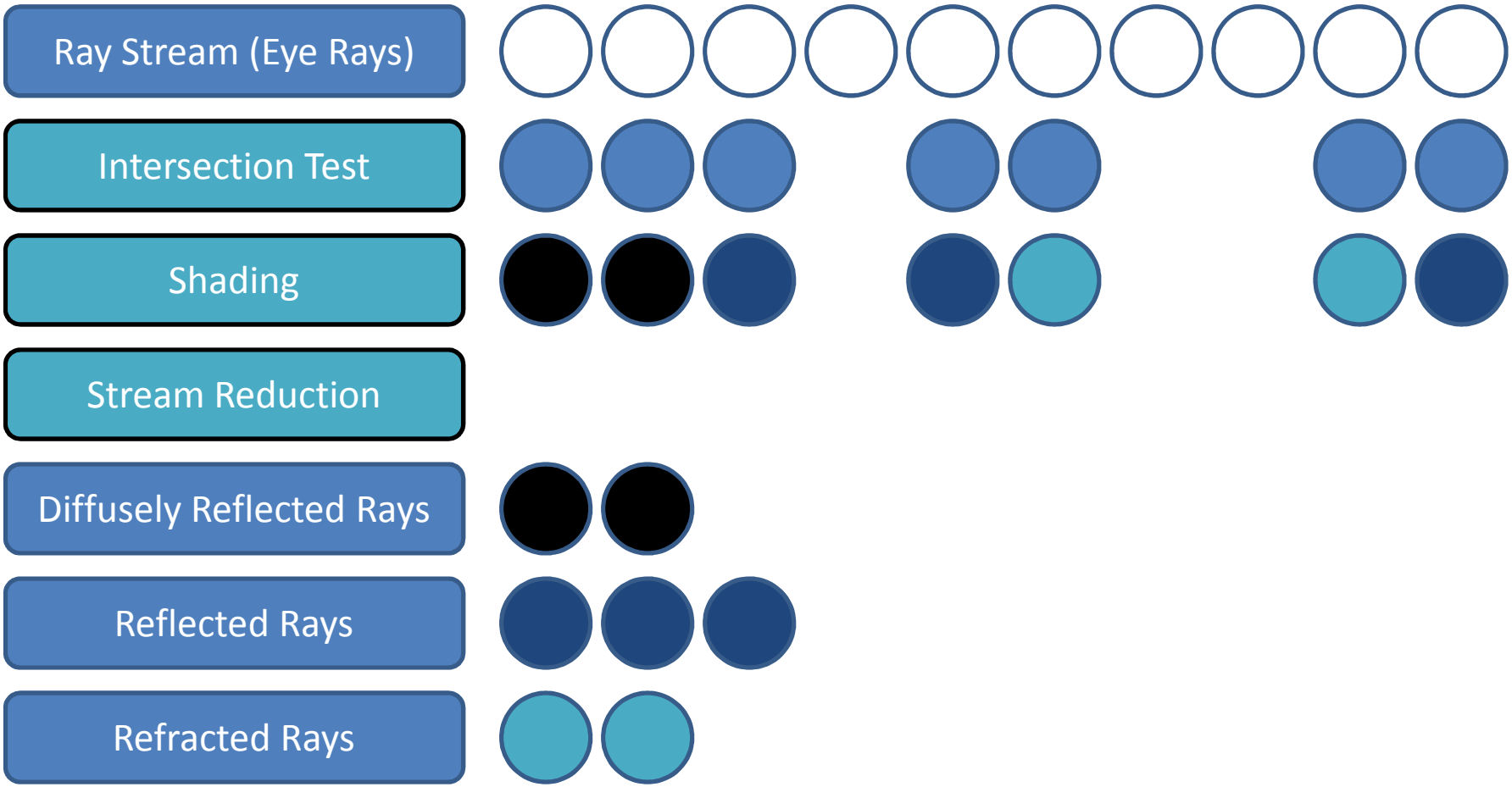
ストリーム・リダクション



ストリーム・リダクション

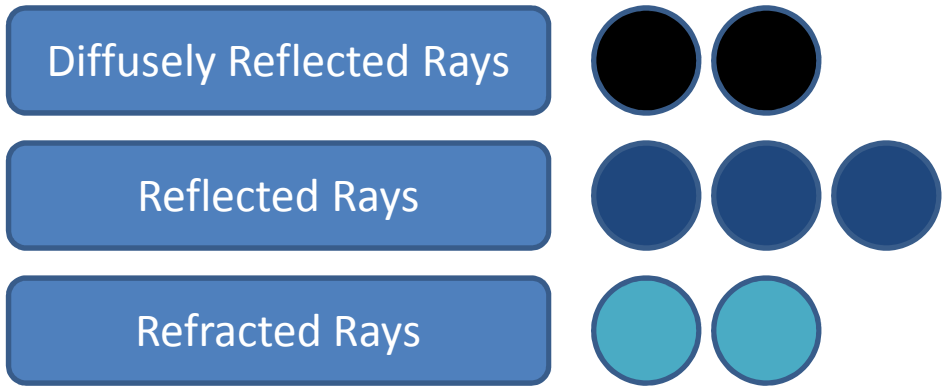


ストリーム・リダクション



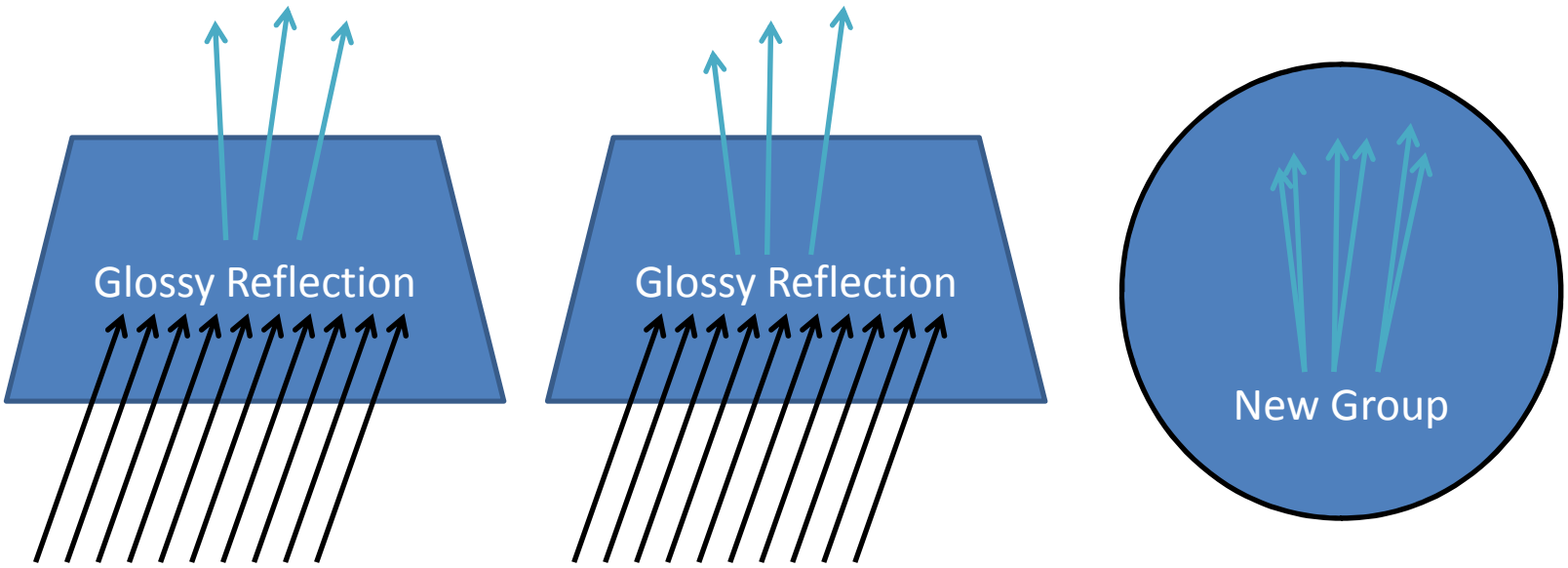
フラグメント・コレクション

- 異なるプロセッサに異なるカーネルをアサインできない
- レイ・ストリームが小さくなりすぎると、無駄が生じる



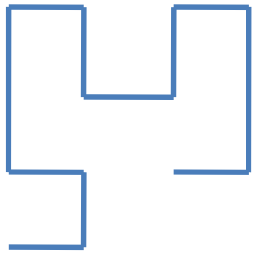
フラグメント・コレクション

- 小さなストリームをフラグメントとして扱う
- ストリーム・プロセッサをビジーに保つため、フラグメントはまとめて一括処理



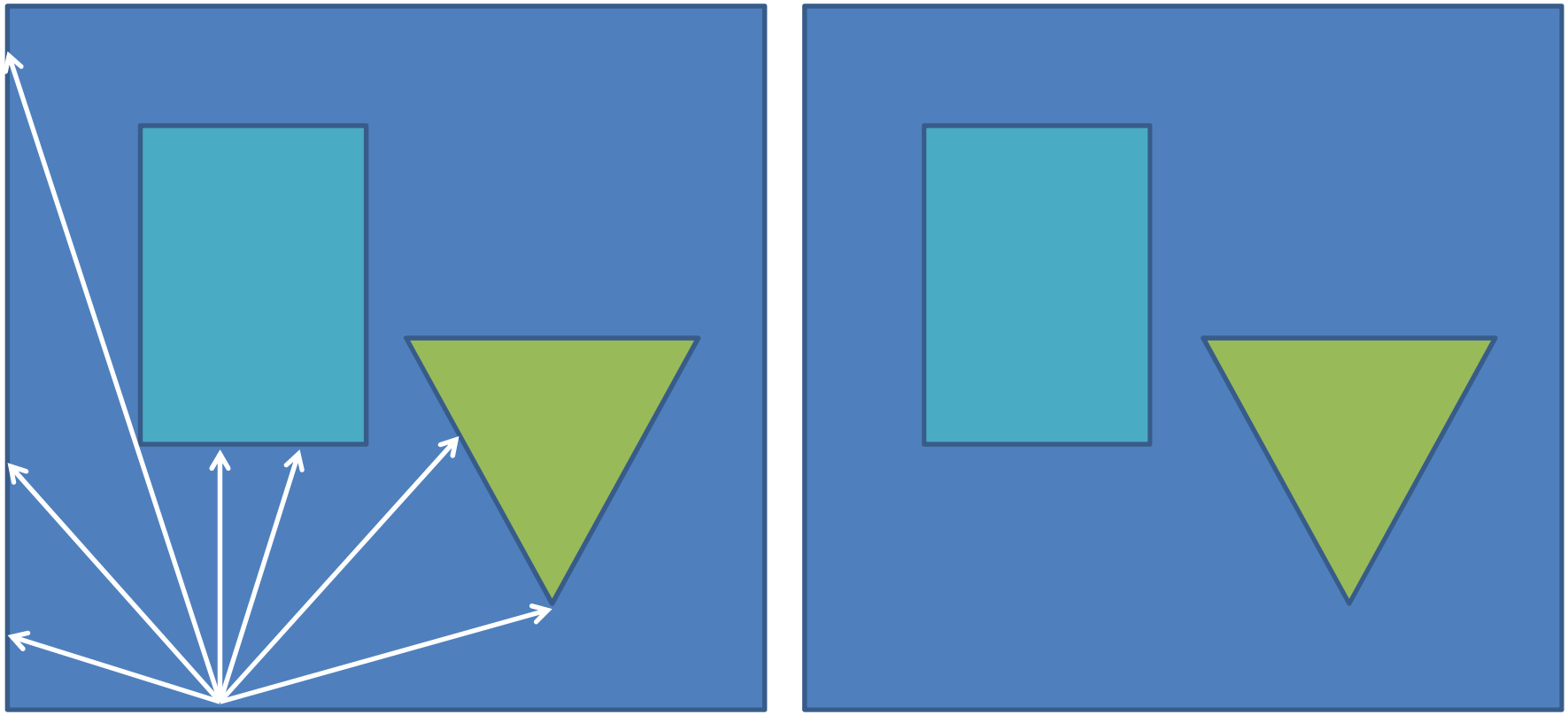
キャッシュ・オブリビアス・データ・レイアウト

- キャッシュ・ミスを減らす“Cache Oblivious Mesh Layout” (Sung-Eui Yoon et al.)
- Vertices, Meshes, Triangle IDs, Photonsなど
- 空間充填曲線が最も簡単な例



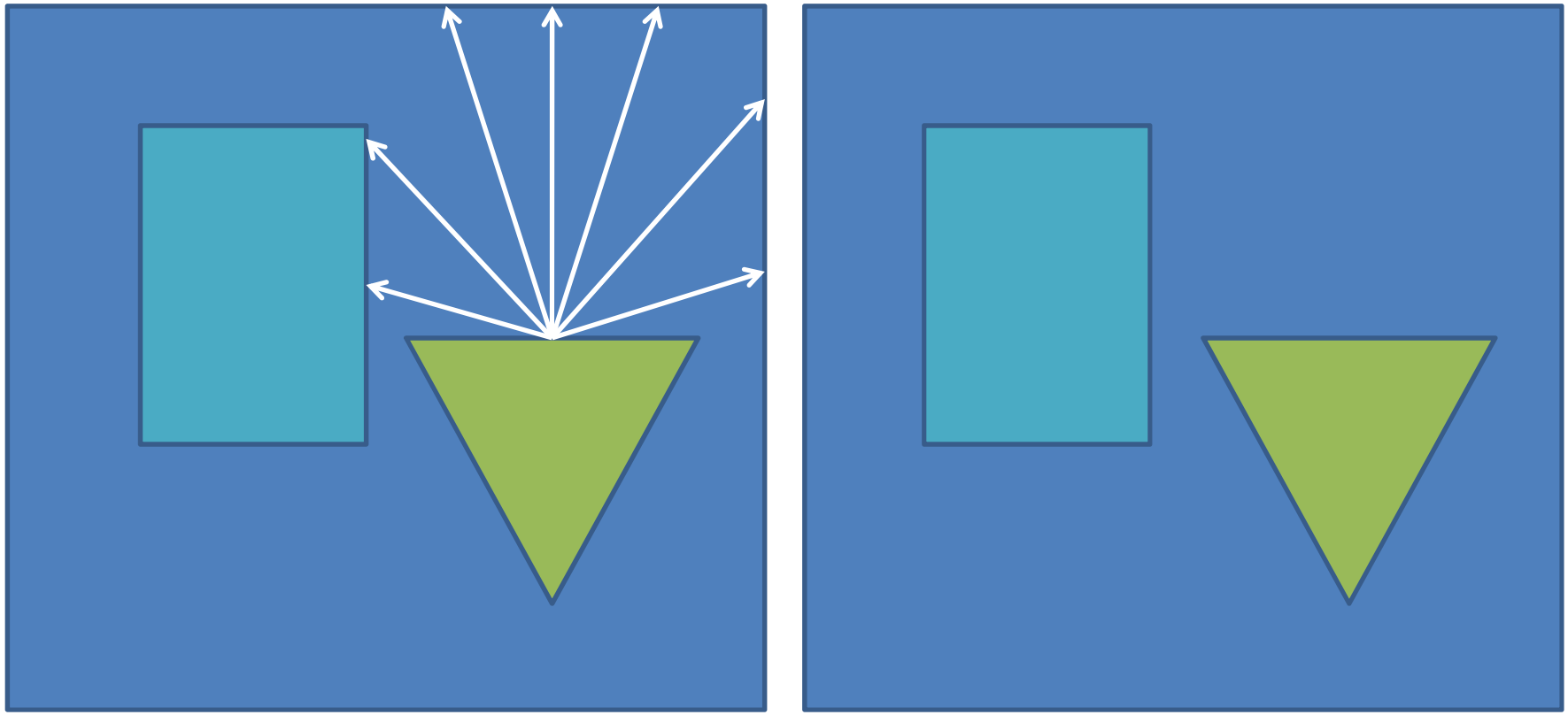
コヒーレント・レイ・トレーシング

- “High-Quality Global Illumination Rendering Using Rasterization” (Hachisuka)



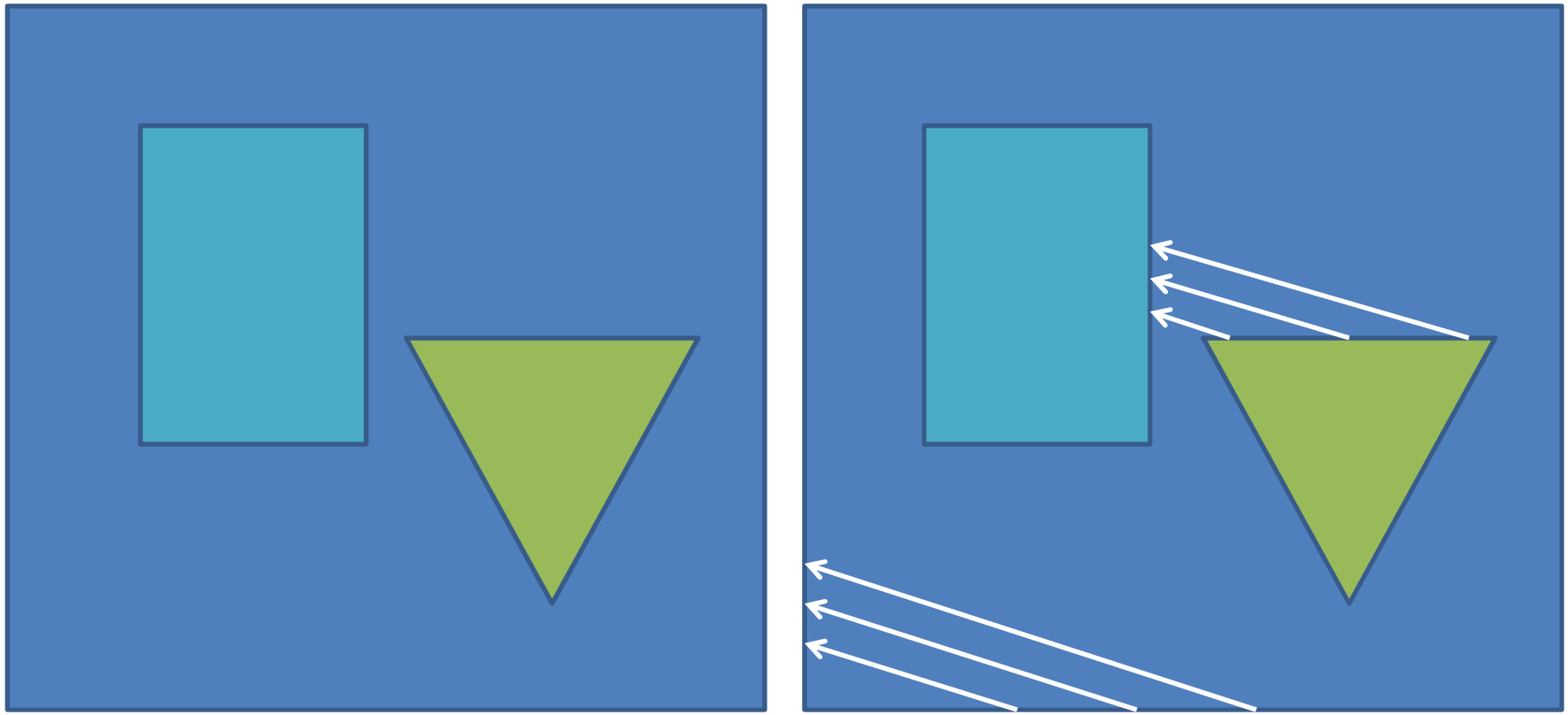
コヒーレント・レイ・トレーシング

- “High-Quality Global Illumination Rendering Using Rasterization” (Hachisuka)



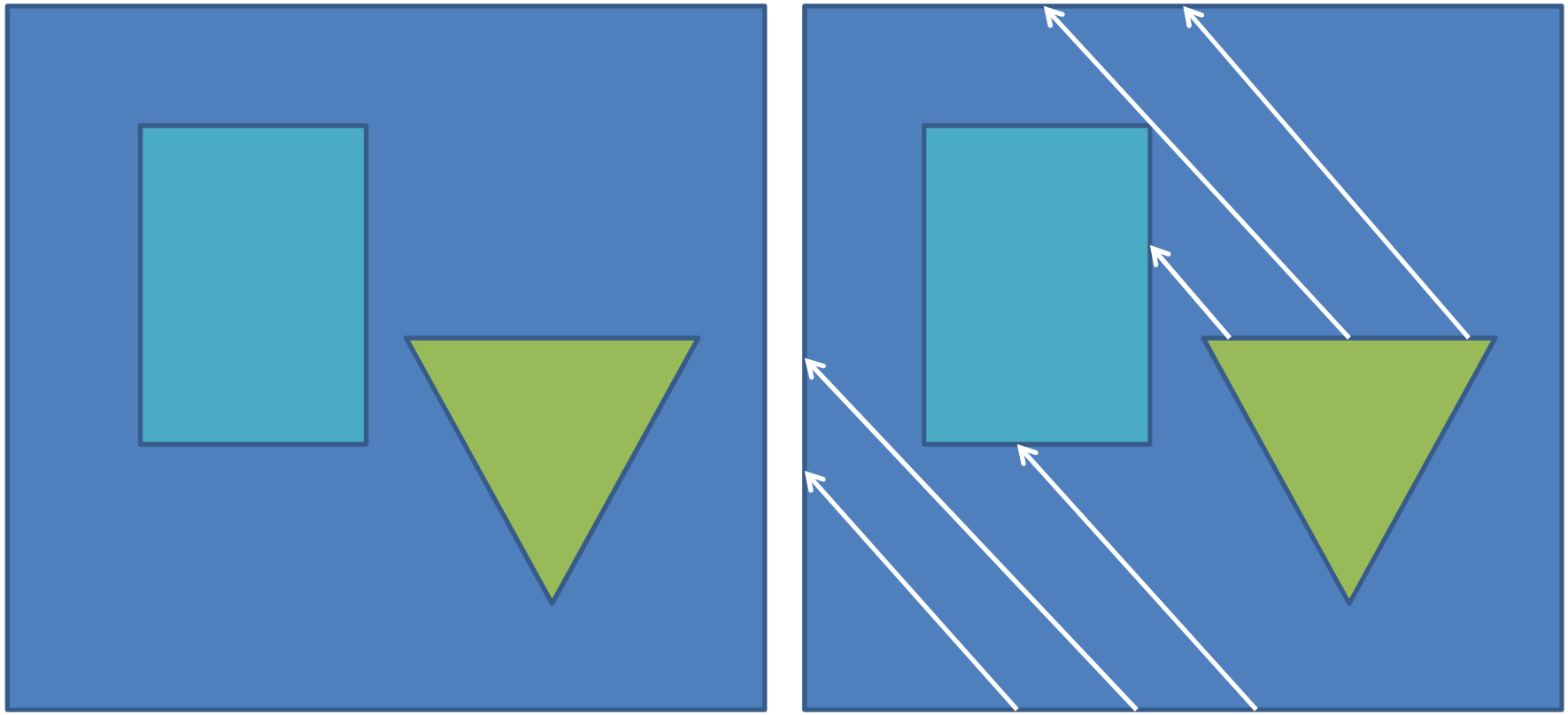
コヒーレント・レイ・トレーシング

- “High-Quality Global Illumination Rendering Using Rasterization” (Hachisuka)



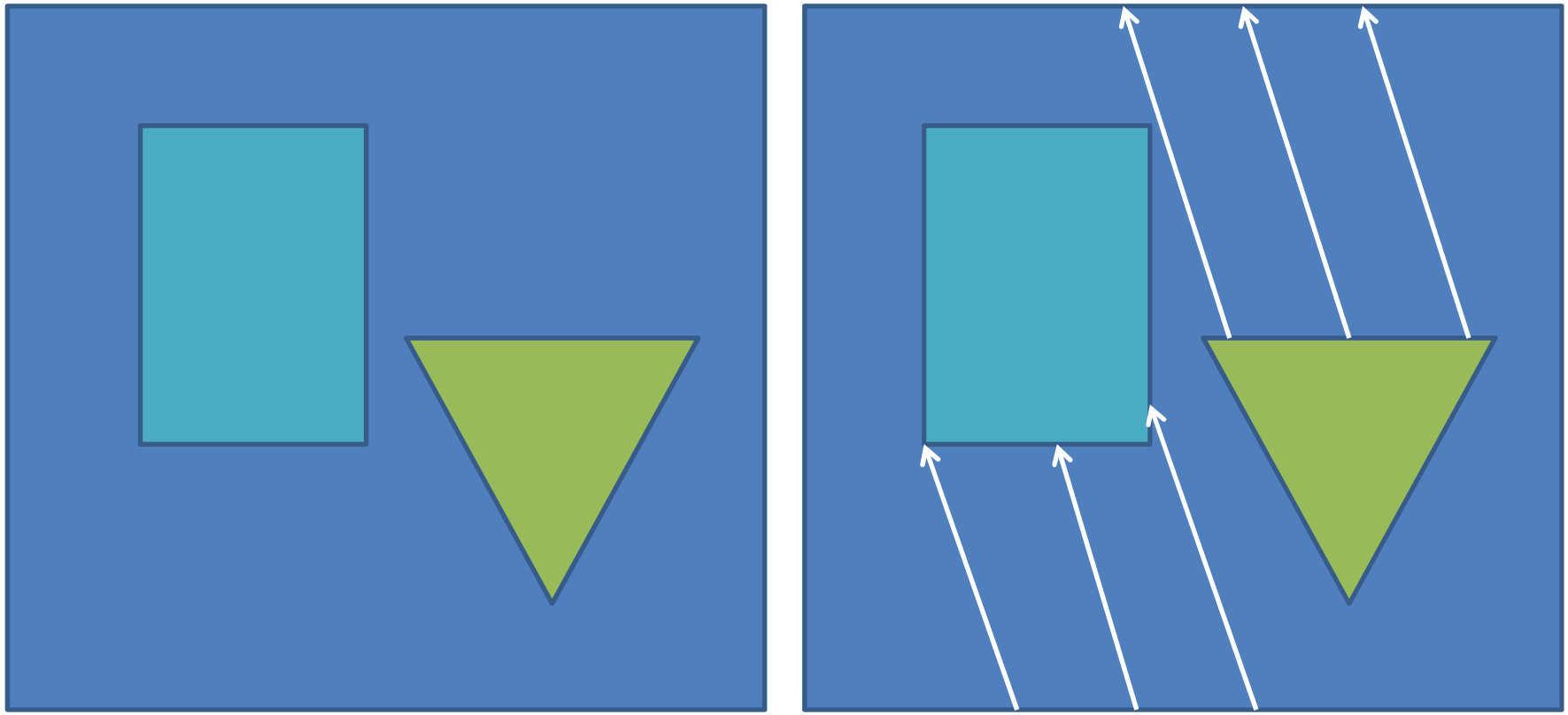
コヒーレント・レイ・トレーシング

- “High-Quality Global Illumination Rendering Using Rasterization” (Hachisuka)



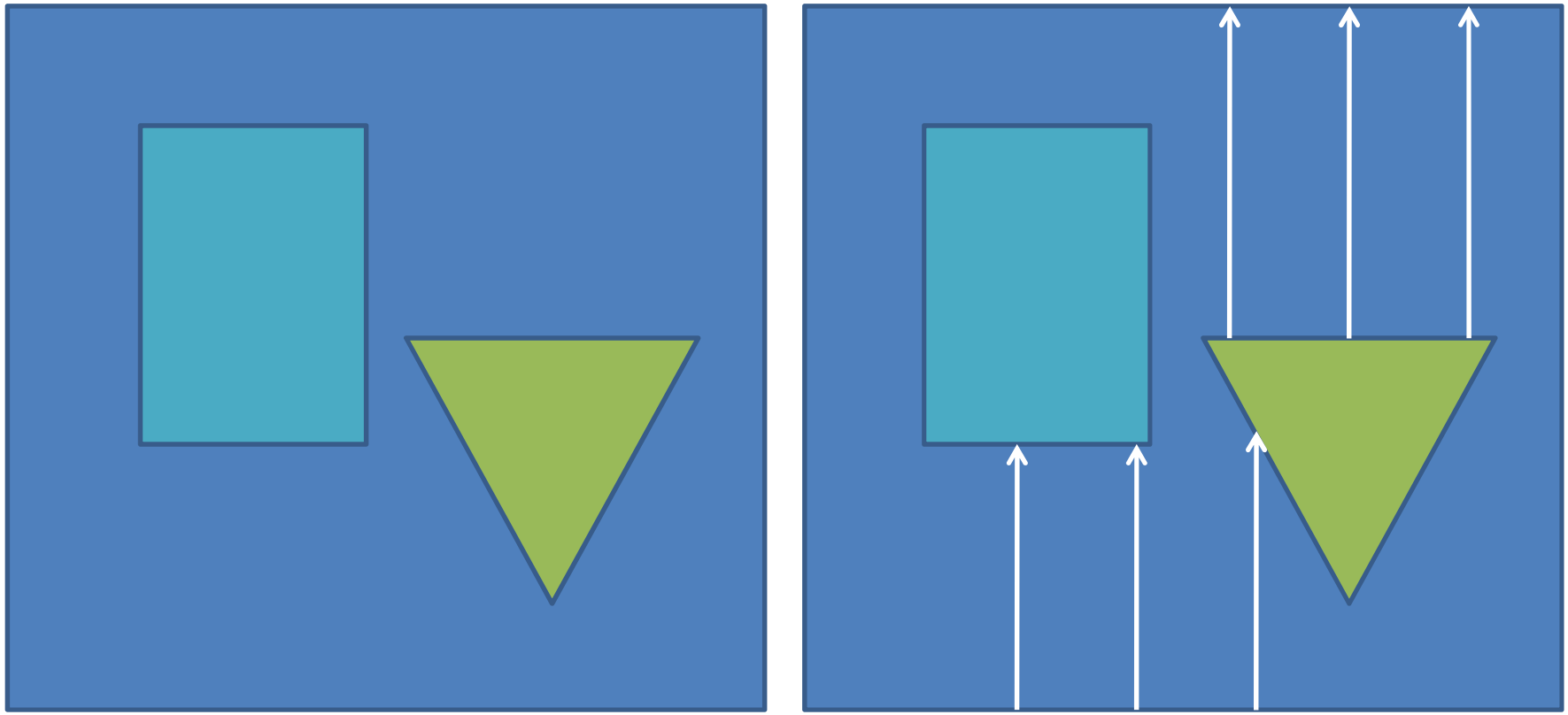
コヒーレント・レイ・トレーシング

- “High-Quality Global Illumination Rendering Using Rasterization” (Hachisuka)



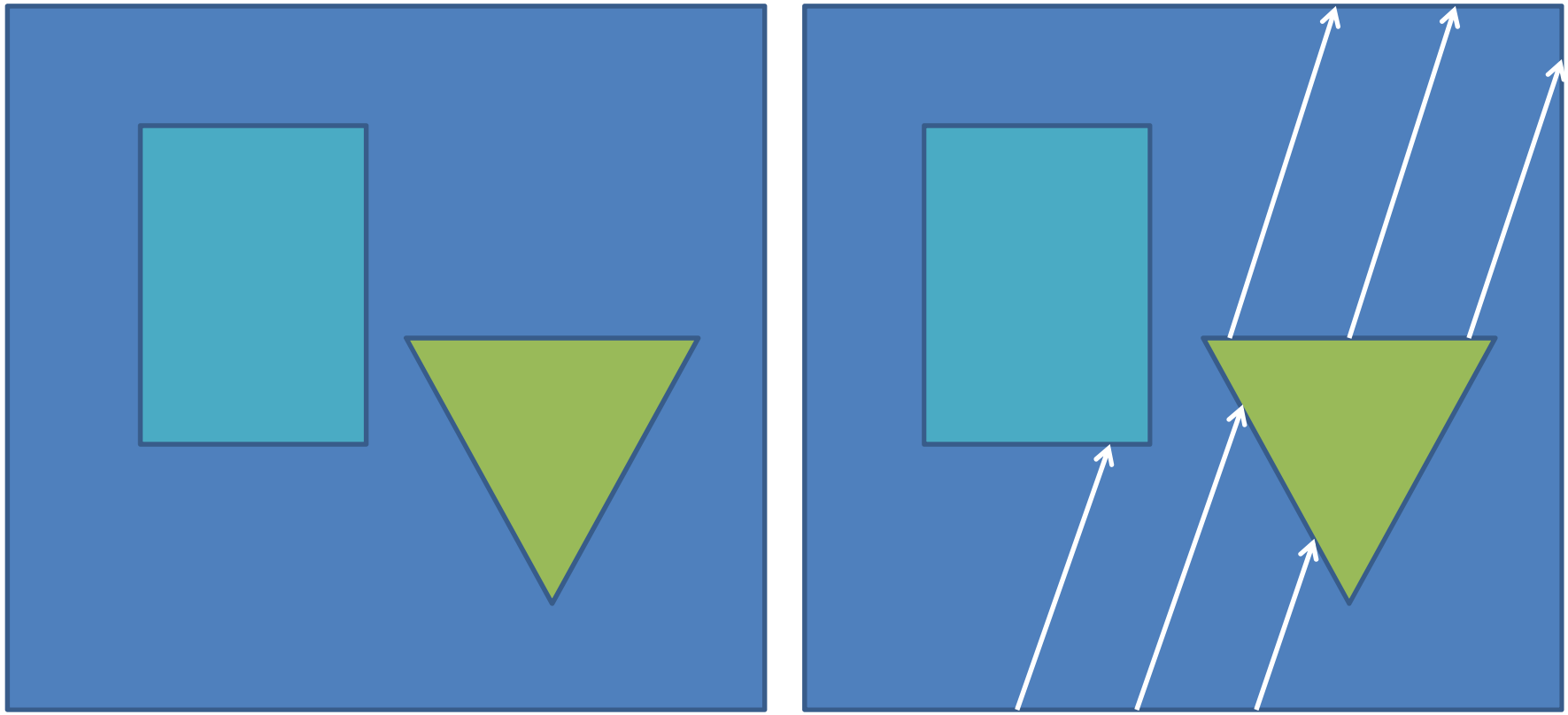
コヒーレント・レイ・トレーシング

- “High-Quality Global Illumination Rendering Using Rasterization” (Hachisuka)



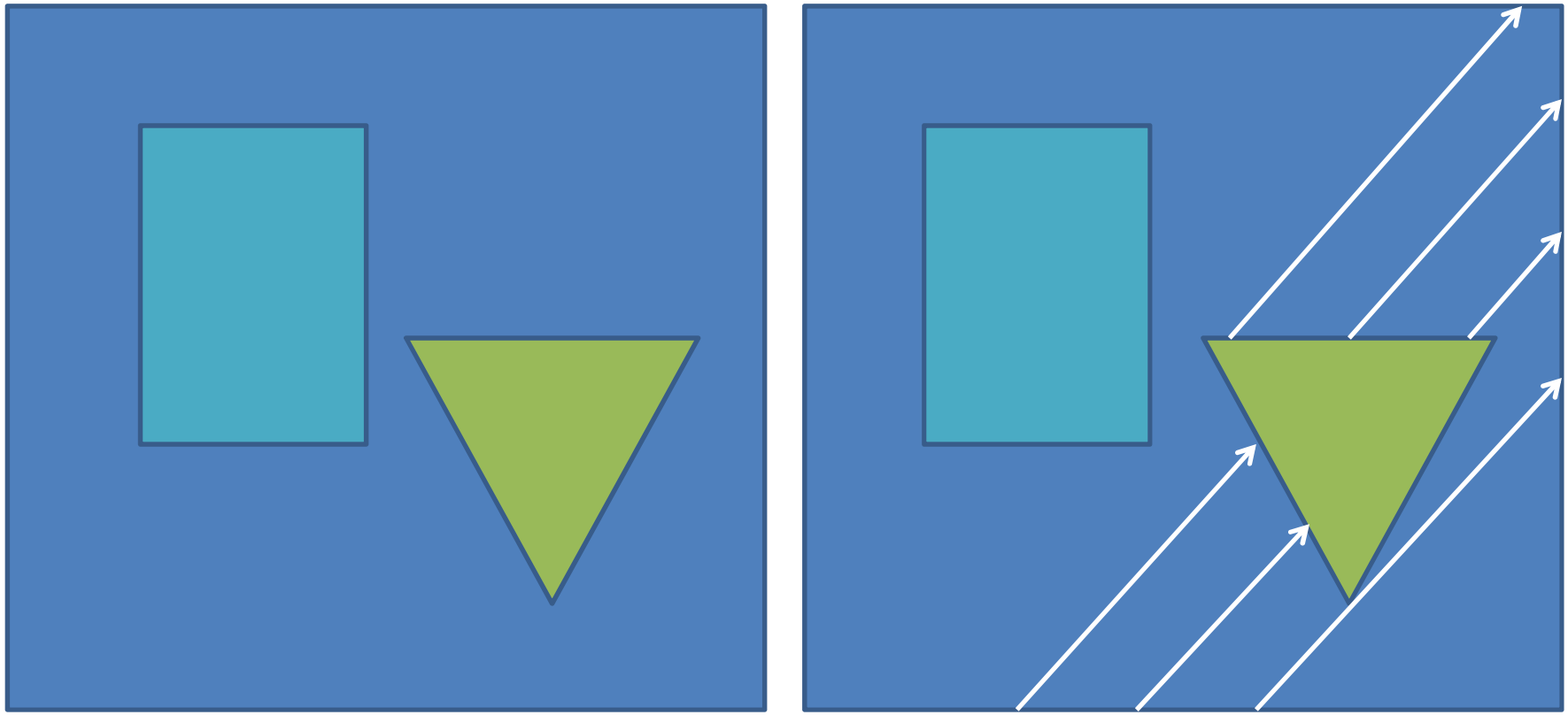
コヒーレント・レイ・トレーシング

- “High-Quality Global Illumination Rendering Using Rasterization” (Hachisuka)



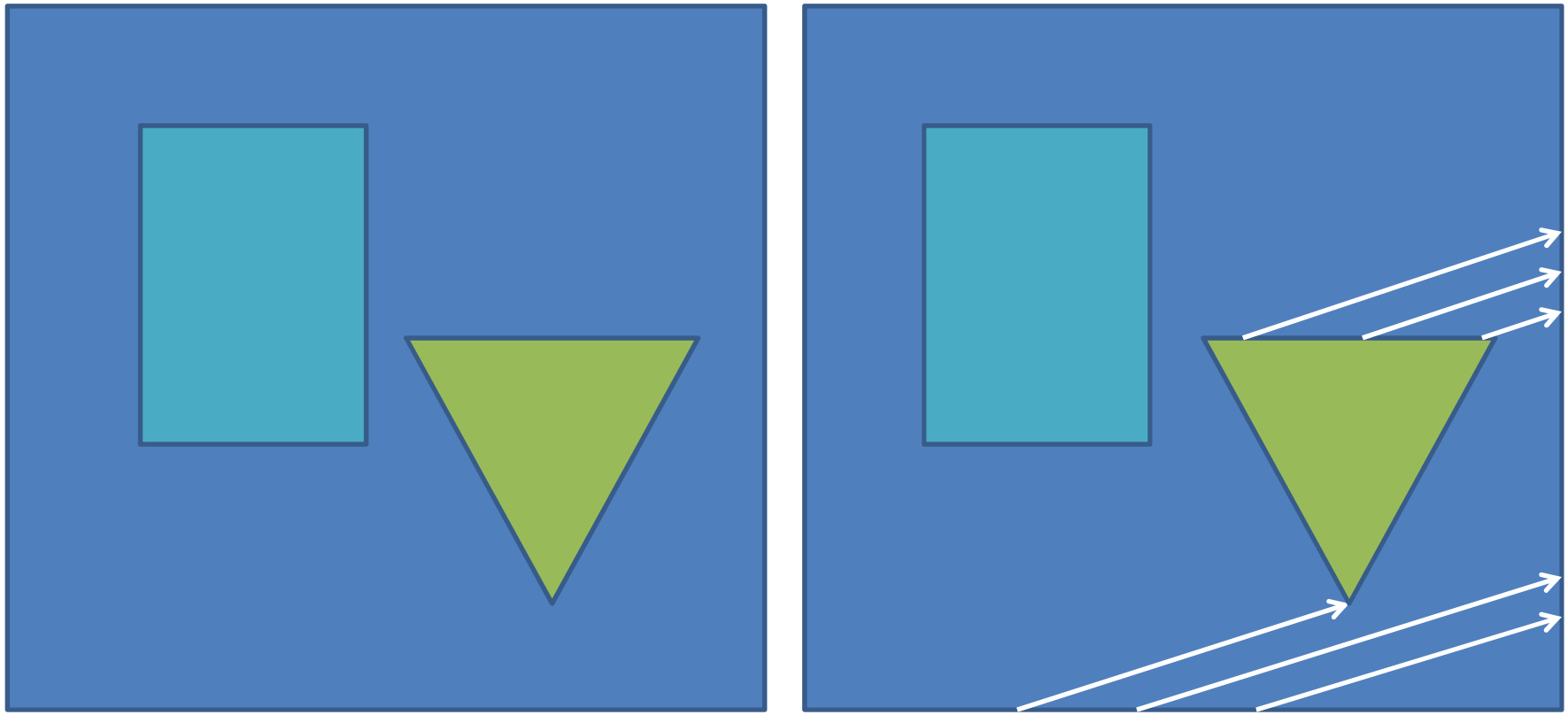
コヒーレント・レイ・トレーシング

- “High-Quality Global Illumination Rendering Using Rasterization” (Hachisuka)



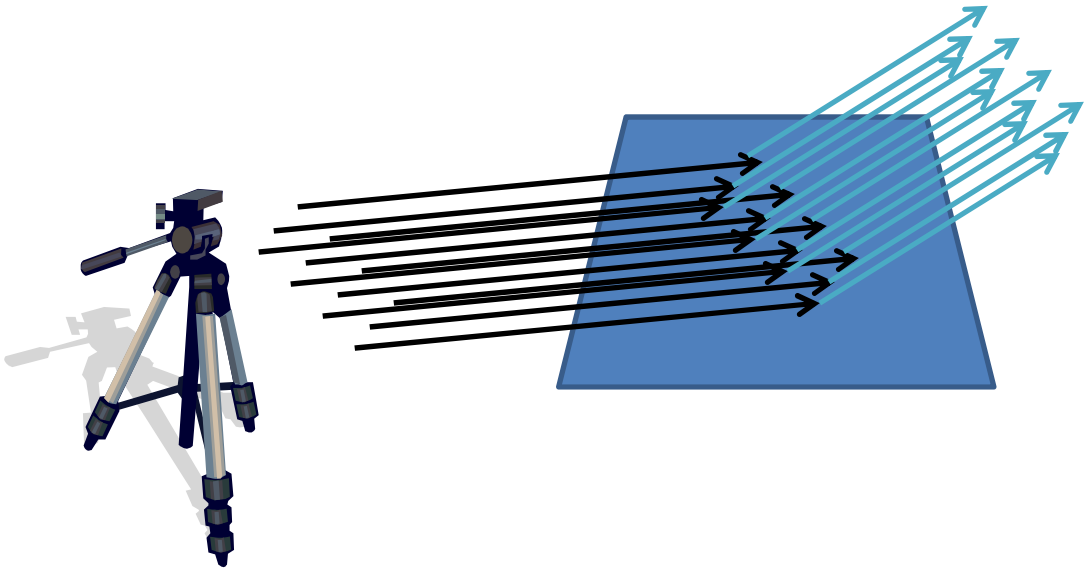
コヒーレント・レイ・トレーシング

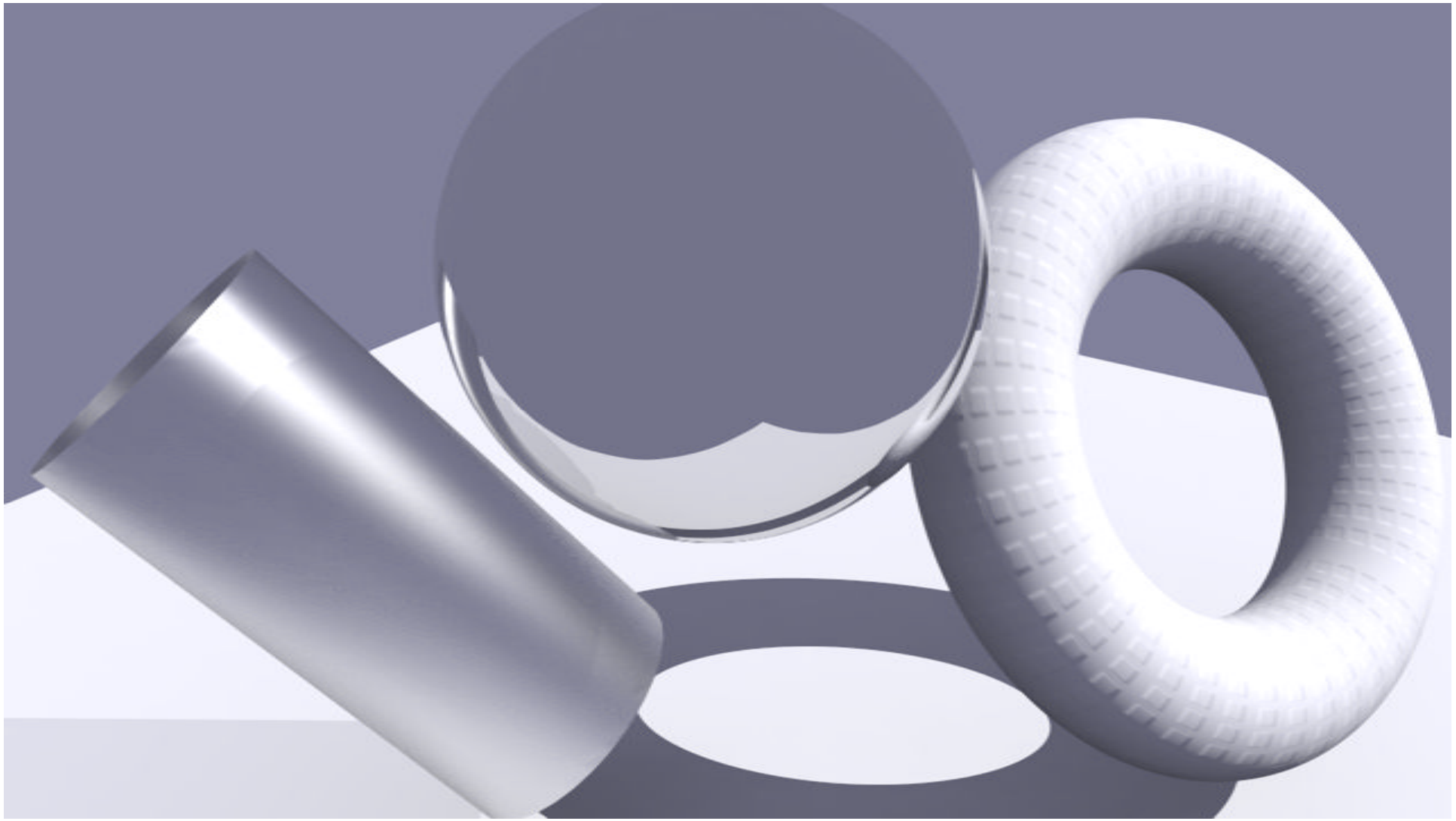
- “High-Quality Global Illumination Rendering Using Rasterization” (Hachisuka)



コヒーレント・レイ・トレーシング

- 1次レイだけでなくファイナル・ギャザー・レイもヒルベルト順で処理する→処理時間 50%ほど短縮





DEMO

音声認識アプローチ

はじめに

ゲーム制作での音声の主な用途

- 発話音声のアライメント情報をリップシンク・データとして利用
- 発話音声を文字情報として提示する音声入力として利用
- 発話音声から特定単語を抽出し音声操作として利用

従来のリップシンクの実際

- FFXIIIでのムービー制作での作業フロー

  ①リップシンクを決める

  ②視線を決める

  ③表情を決める

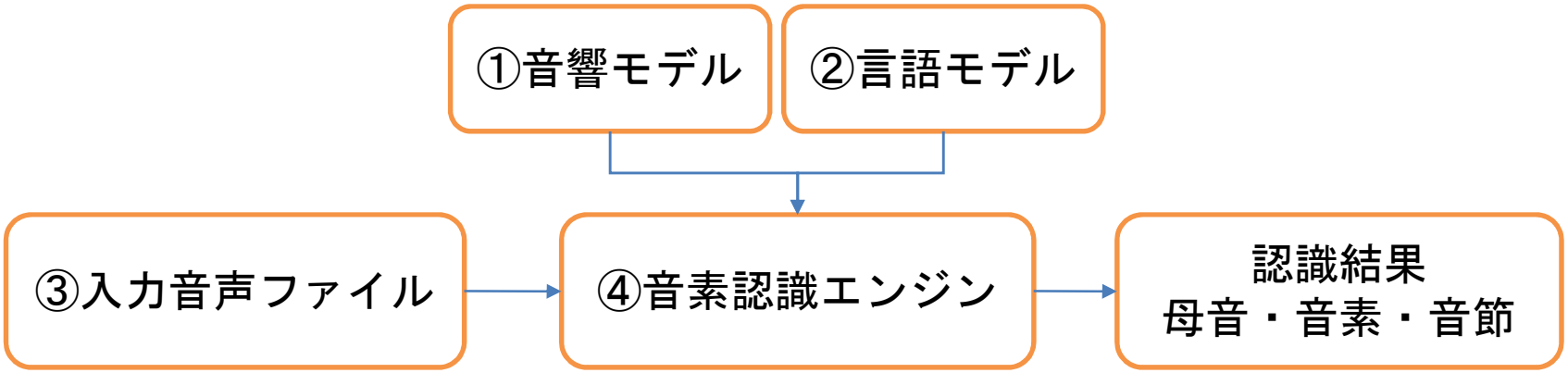
優秀なアニメーターが常時3名でDCCツール上で1秒当たり0.6人日分のコストを掛けていた

今回の目的

・音声認識でリップシンクの自動化を

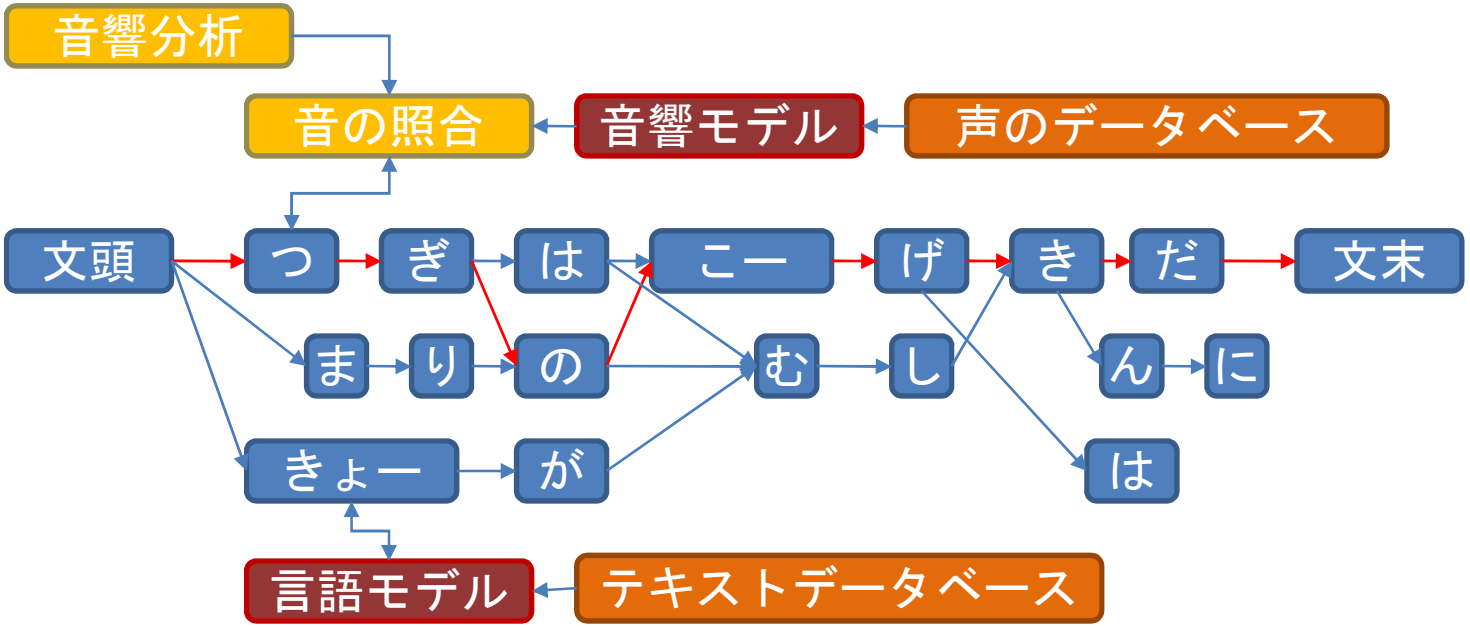
①コストの削減（人・モノ・時間）
②品質の向上（カットシーン，NPC）
③品質の維持（デザイナースキルでのバラつき無くす）

音素認識とは



① 音響モデル（HMM各音素の周波数特徴の分布統計量）
② 言語モデル（音節連鎖統計量で音節バイグラムモデル）
③ 音声ファイル
④ 与えられた音に対して音響スコアと言語スコアの総和を求める

音声認識とは



与えられた音声に対して、どの経路を通った時に音響的かつ言語的に最も確率が高いのかを計算し、その経路を認識結果とする

音声認識率は１００％ではない

対処方法の選択肢

- 全台詞で共通の音響モデルを作る

  （音響学習の手間は省けるが、認識率は微増）

- 声優さん毎の台詞で音響モデルを作る

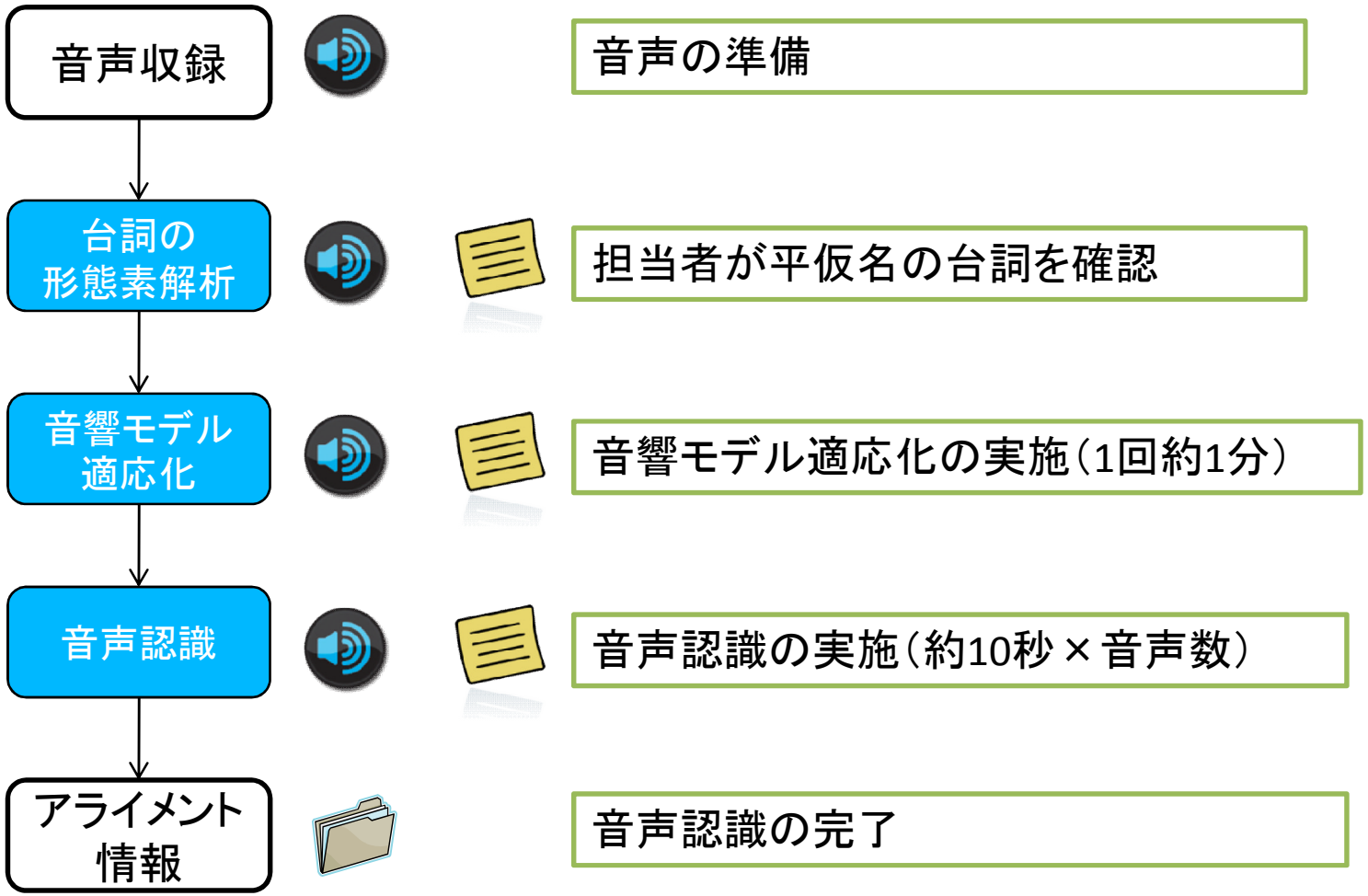
  （実際の台詞で実施することで認識率は上がる）

- 台詞毎に音響モデルを作る

  （音響モデル学習時間は増えるが、認識率は100%）

リップシンクの場合はアライメント情報（音素種と時間）を得ることが目的で、音響モデルの作成時間を抑制することも考慮して、声優さん毎の音響モデルを1つ作ることとした

作業フローの設計

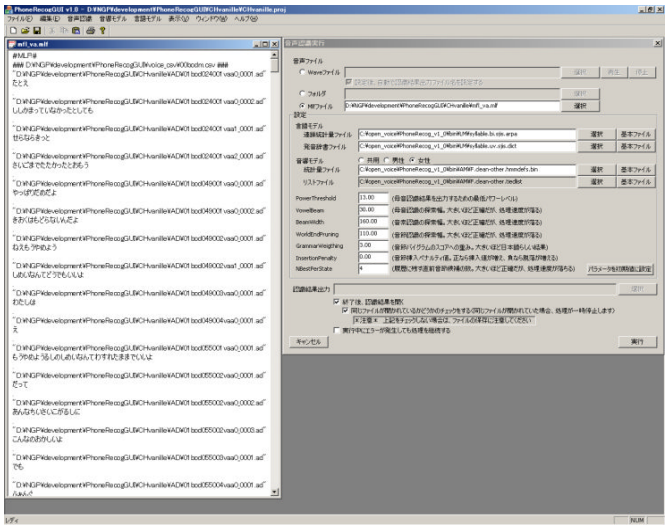


開発

- リップシンク制作フローの構築

NHK放送技術研究所の音声認識をライセンス

- 音声認識エンジン技術の提供
- 音響モデルの適応化方法の開示



当社・研究開発部での開発

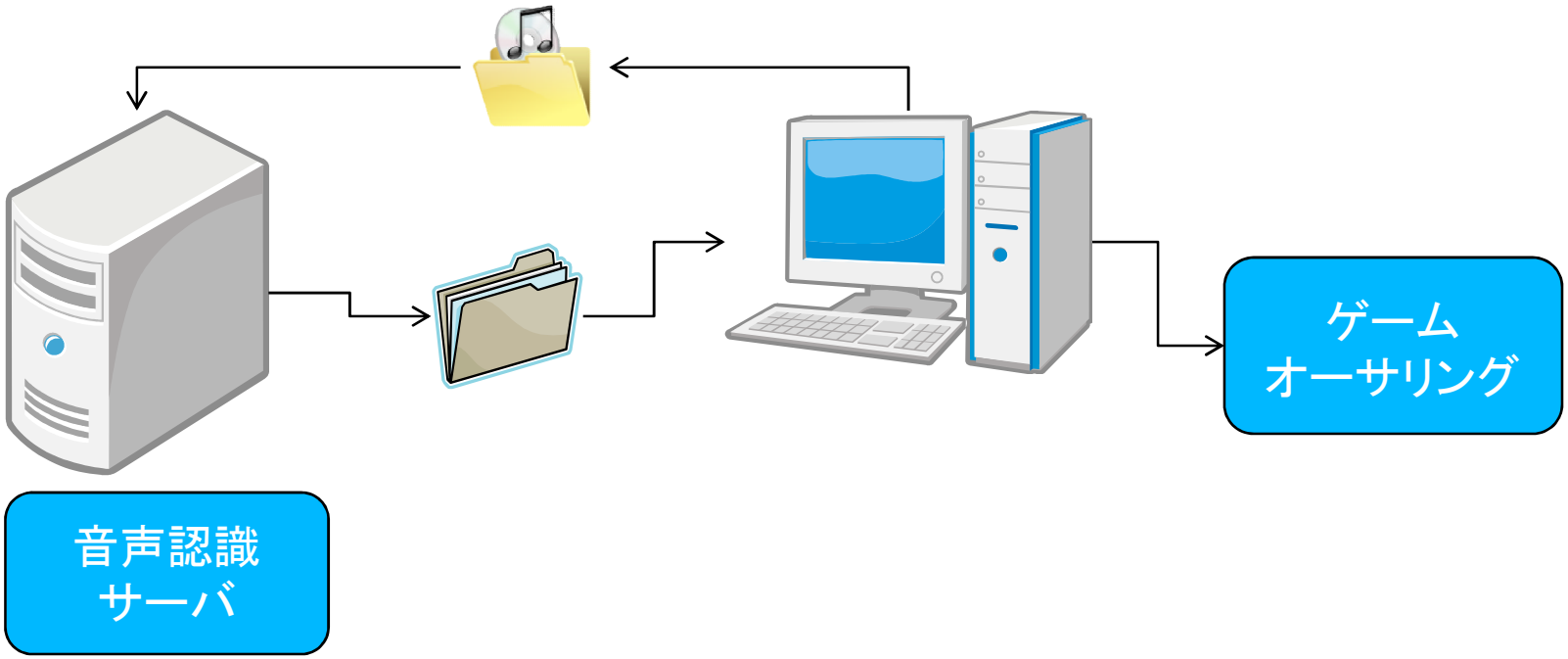
- 音声認識エンジンのAPI化
- 音響/言語モデル学習ツール開発
- 上記の要素技術を用いたシステム化

音声認識エンジンの選定理由

- 言語モデルが音節バイグラムモデルであり、即時応答性能が求められる場合（逐次型）への対応の可能性がある
- 標準男性，女性,汎用の基本音響モデルの精度が高い
- 音響モデルの適応化が容易である
- 言語モデルの学習が容易である
- アライメント情報の取得部分の技術開示があり、出力結果の情報量が多い
- カスタマイズ開発でき、且つ包括ライセンスに対応可能であった

運用形態

・クライアント＆サーバ環境として利用



DEMO

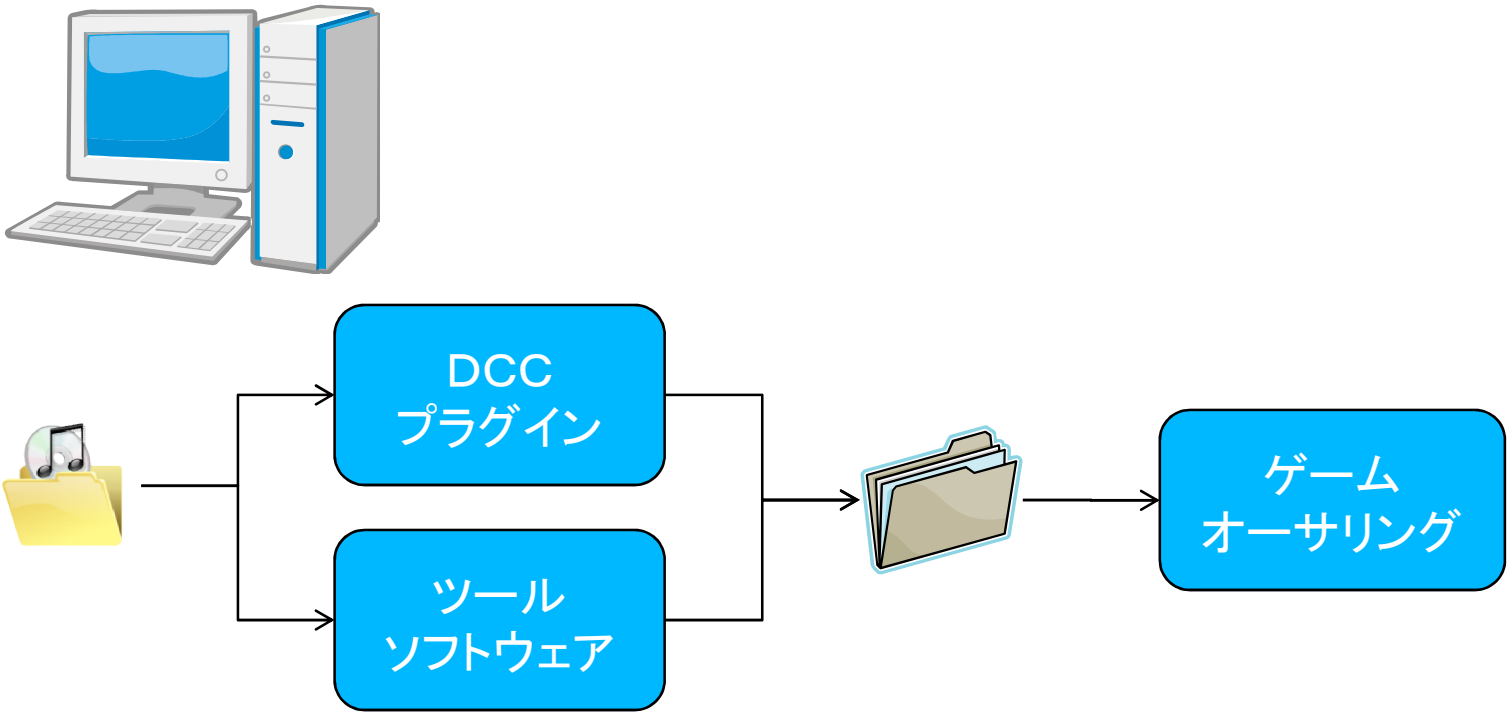
実績

音声認識技術を実際の制作で利用

- NPCのリップシンク（カットシーン）制作で利用したところ従来手法と比べ1/8のコスト削減効果が得られた
- スキルの高いアニメータは主要キャラクターに注力できるという副次的な効果も

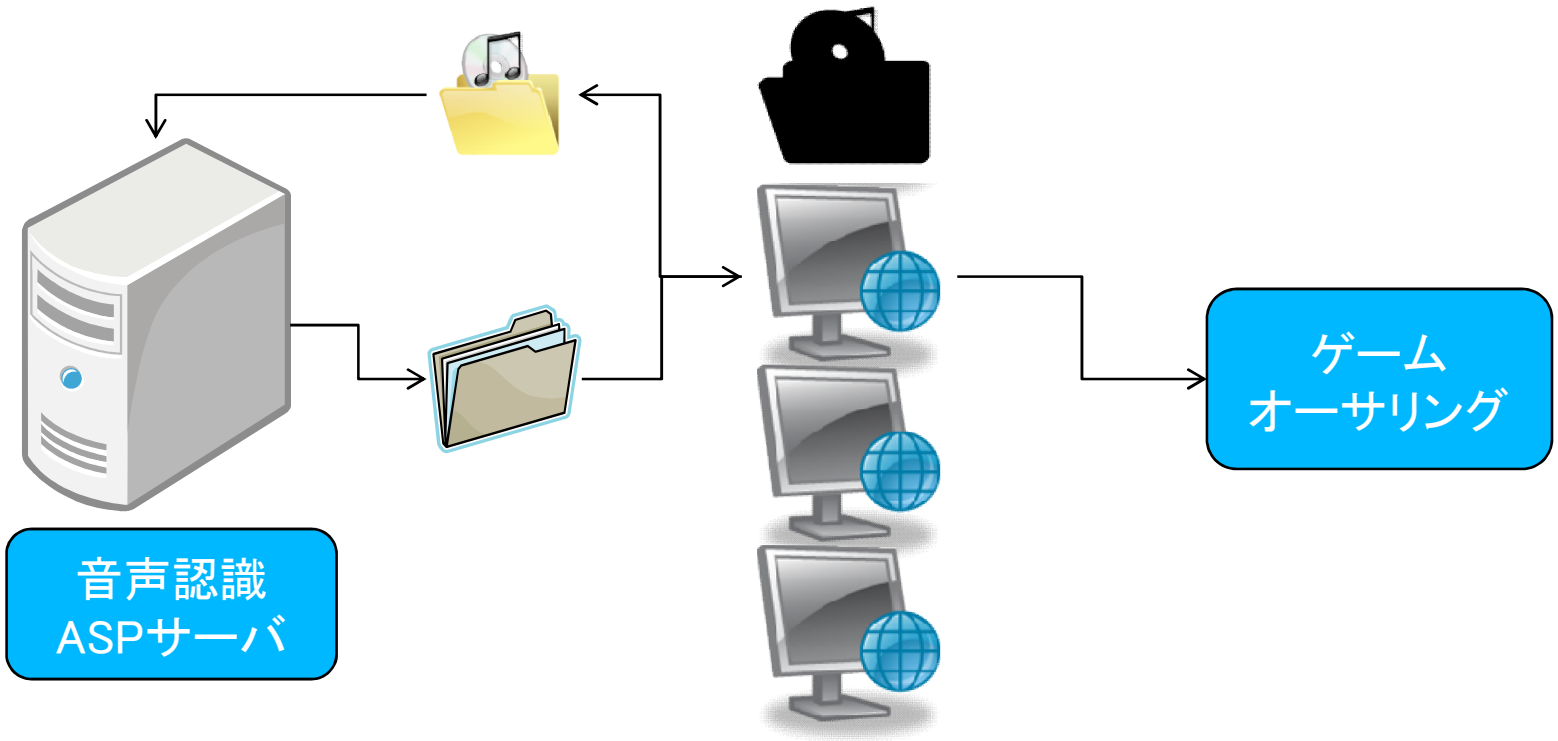
運用形態2

・スタンドアローン・アプリケーションとして利用



運用形態3

・音声認識ASPサーバ環境として利用



今後の課題

- 音声認識技術の利用機会を増やすこと
- 次世代の制作フローへ適応させること
- 他の技術を融合して表現力を上げること
- 処理時間を短縮すること

御協力

SQUARE ENIX

Q&A

ご静聴有難う御座いました。